

**305-EMD-100**

**EOSDIS Maintenance and Development Project**

**Release 7.20 Segment/Design  
Specifications  
for the EMD Project**

Revision --

July 2007

Raytheon Company  
Upper Marlboro, Maryland

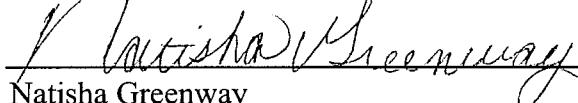
**Release 7.20  
Segment/Design Specifications  
for the EMD Project**

**Revision --**

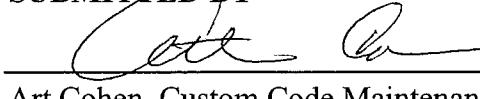
**July 2007**

Prepared Under Contract NAS5-03098  
CDRL Item #023

**RESPONSIBLE ENGINEER**

Natisha Greenway July 26, 2007  
Natisha Greenway Date  
EOSDIS Maintenance and Development Project

**SUBMITTED BY**

Art Cohen July 26, 2007  
Art Cohen, Custom Code Maintenance Date  
EOSDIS Maintenance and Development Project

**Raytheon Company**  
Upper Marlboro, Maryland

This page intentionally left blank.

# Preface

---

This document is a formal contract deliverable. It requires Government review and approval within 45 business days. Changes to this document will be made by document change notice (DCN) or by complete revision.

Any questions or proposed changes can be addressed to:

Data Management Office  
The EMD Project Office  
Raytheon Company  
1616 McCormick Drive  
Upper Marlboro, MD 20774-5301

## Revision History

Document Number	Status/Issue	Publication Date	CCR Number
305-EMD-100	Original	July 2007	07-0332

This page intentionally left blank.

## Abstract

---

The Release 7.20 Segment/Design Specification is an overview description of the EMD Project. The functionality of the ECS software is described at the Subsystem, Computer Software Configuration Item (CSCI), Computer Software Component (CSC), and Process levels. Architecture and context diagrams illustrate the process interconnections within the ECS CSCIs and the external connections to other CSCIs, subsystems, and specified segment interfaces. Interface event description tables describe the data, messages, notifications, or status information that occurs at each level of functionality within the ECS. A basic description of the Commercial Off The Shelf (COTS) software and hardware used in ECS is included.

The high-level design in this document is the level of information derived from requirement sources, and used by the development team to complete the ECS design implementation for a software system at a 7.20 state of maturity.

**Keywords:** Release 7.20, Overview, SDPS, CSMS, Design, Detailed Design, Subsystem, Architecture, Software, Hardware, Object Oriented, Security, Gateway, System Management, Reports, User Interface and GUI.

This page intentionally left blank.

# **Contents**

---

## **1. Introduction**

1.1	Purpose and Scope .....	1-1
1.2	Document Organization .....	1-1

## **2. Related Documentation**

2.1	Parent Documents .....	2-1
2.2	Applicable Documents.....	2-1
2.2.1	Other Related Documents and Documentation.....	2-1
2.3	Information Documents Not Referenced.....	2-2
2.4	ECS Tool Descriptions .....	2-3
2.4.1	Rational Rose.....	2-3
2.4.2	ClearCase Baseline Manager Configuration Management Tool .....	2-4

## **3. System Description**

3.1	Mission and Release 7.20 Objectives .....	3-1
3.1.1	Release 7.20 Capabilities .....	3-1
3.2	Release 7.20 Architecture Overview .....	3-9
3.2.1	Release 7.20 Context Description.....	3-12
3.2.2	Release 7.20 Architecture .....	3-14

## **4. Subsystem Description**

4.1	Data Server Subsystem Overview .....	4-19
4.1.1	Science Data Server Software Description.....	4-28
4.1.2	Data Distribution Software Description (REMOVED) .....	4-50
4.2	DPL Ingest Subsystem Overview .....	4-51
4.2.1	DPL Ingest Computer Software Configuration Item Description .....	4-53

4.3	Client Subsystem Overview.....	4-63
4.3.1	Desktop Software Description .....	4-66
4.3.2	SSI&T Tools Description .....	4-72
4.4	Data Management Subsystem Overview .....	4-73
4.4.1	Data Dictionary Software Description.....	4-81
4.4.2	V0 Gateway Software Description .....	4-91
4.4.3	Data Management Subsystem Hardware .....	4-110
4.5	Order Manager Subsystem Overview.....	4-111
4.5.1	Order Manager Subsystem Software Description .....	4-114
4.6	Planning Subsystem Overview (REMOVED).....	4-128
4.7	Data Processing Subsystem Overview (REMOVED) .....	4-129
4.8	Communications Subsystem Overview .....	4-130
4.8.1	The Distributed Computing Configuration Item Software Description .....	4-138
4.8.2	The Distributed Computing Configuration Item Context.....	4-182
4.8.3	Distributed Computing Configuration Item Architecture.....	4-191
4.8.4	Distributed Computing Configuration Item Process Descriptions .....	4-192
4.8.5	Distributed Computing Configuration Item Process Interface Descriptions ...	4-192
4.8.6	Distributed Computing Configuration Item Data Stores .....	4-193
4.8.7	Communications Subsystem Hardware CI Description .....	4-193
4.9	System Management Subsystem Overview.....	4-194
4.9.1	Management Software Computer Software Configuration Item Description .	4-203
4.9.2	Management Logistics Computer Software Configuration Item Description .	4-257
4.9.3	ECS Assistant Script Library .....	4-266
4.9.4	Systems Management Subsystem Hardware Components .....	4-278
4.10	Internetworking Subsystem (ISS) Overview .....	4-280
4.10.1	Internetworking Subsystem Description.....	4-281
4.10.2	Network COTS Hardware.....	4-283
4.11	EMD General Process Failure Recovery Concepts .....	4-286
4.11.1	Client-Server Rebinding .....	4-286
4.11.2	Sybase Reconnecting .....	4-287
4.11.3	Request Identification .....	4-288
4.11.4	Senior Clients.....	4-288
4.11.5	Request Responsibility .....	4-289

4.11.6	Queues.....	4-290
4.11.7	Request Responses.....	4-290
4.11.8	Duplicate Request Detection .....	4-294
4.11.9	Server Crash and Restart.....	4-295
4.11.10	Client Crash and Restart.....	4-297
4.12	Product Distribution System (PDS) Subsystem Overview (REMOVED) .....	4-301
4.13	Spatial Subscription Server (SSS) Subsystem Overview .....	4-302
	4.13.1 Spatial Subscription Server Architecture.....	4-303
4.14	Data Pool Subsystem Overview.....	4-310
	4.14.1 Data Pool Subsystem Context.....	4-312
	4.14.2 Data Pool Hardware Context .....	4-314
	4.14.3 Data Pool Insert CSCI Functional Overview.....	4-315
	4.14.4 WebAccess CSCI Functional Overview.....	4-322
	4.14.5 Data Stores.....	4-334
4.15	Bulk Metadata Generation Tool Subsystem Overview .....	4-335
	4.15.1 BMGT Architecture .....	4-338
	4.15.2 The BMGT Subsystem Software Description .....	4-341
4.16	OGC-ECHO Adaptor (OEA) Subsystem Overview .....	4-365
	4.16.1 OGC-ECHO Adaptor Architecture.....	4-367
4.17	Open Geospatial Consortium (OGC) Web Services (OWS) Functional Overview (REMOVED).....	4-372

## **5. Limitations of Current Implementation**

5.1	Data Server Subsystem .....	5-1
5.2	Planning Subsystem (REMOVED).....	5-1
5.3	Infrastructure Subsystem .....	5-1
5.4	MSS Subsystem .....	5-1
5.5	Product Distribution System (PDS) Subsystem (REMOVED) .....	5-2
5.6	Spatial Subscription Server (SSS) Subsystem .....	5-2
5.7	Open Geospatial Consortium (OGC) Web Services (OWS) Subsystem (REMOVED) .....	5-2
5.8	Order Manager Server (OMS) Subsystem.....	5-2

5.9	Data Management (DMS) Subsystem .....	5-2
5.10	Data Pool Ingest (DPLINGEST) Subsystem .....	5-2

## List of Figures

Figure 3.2-1.	Example Hierarchical Software Diagram .....	3-11
Figure 3.2-2.	Release 7.20 Context Diagram.....	3-13
Figure 3.2-3.	Subsystem Architecture Diagram .....	3-15
Figure 4.1-1.	Data Server Subsystem Context Diagram.....	4-20
Figure 4.1-2.	SDSRV CSCI Context Diagram .....	4-29
Figure 4.1-3.	SDSRV CSCI Architecture Diagram .....	4-33
Figure 4.2-1.	DPL Ingest Subsystem Context Diagram .....	4-51
Figure 4.2-2.	DPL Ingest CSCI Context Diagram.....	4-54
Figure 4.2-3.	DPL Ingest CSCI Architecture Diagram .....	4-57
Figure 4.3-1.	Client Subsystem Context Diagram.....	4-63
Figure 4.3-2.	DESKT CSCI Context Diagram .....	4-66
Figure 4.3-3.	DESKT CSCI Architecture Diagram .....	4-68
Figure 4.4-1.	Data Management Subsystem Context Diagram .....	4-74
Figure 4.4-2.	Data Dictionary CSCI Context Diagram .....	4-82
Figure 4.4-3.	Data Dictionary CSCI Architecture Diagram .....	4-84
Figure 4.4-4.	V0 GTWAY CSCI Context Diagram .....	4-92
Figure 4.4-5.	V0 GTWAY CSCI Architecture Diagram .....	4-97
Figure 4.5-1.	Order Manager Subsystem Context Diagram .....	4-112
Figure 4.5-2.	Order Manager Server CSCI Context Diagram .....	4-115
Figure 4.5-3.	Order Manager Server CSCI Architecture Diagram.....	4-117
Figure 4.5-4.	Production Module CSCI Context Diagram .....	4-123
Figure 4.5-5.	Production Module CSCI Architecture Diagram.....	4-125
Figure 4.8-1.	Communications Subsystem (CSS) Context Diagram.....	4-131
Figure 4.8-2.	Configuration Registry Server Context Diagram.....	4-142

Figure 4.8-3. Configuration Registry Server Architecture Diagram .....	4-144
Figure 4.8-4. Machine-to-Machine Gateway Server Context Diagram.....	4-150
Figure 4.8-5. Machine-to-Machine Gateway Server Architecture Diagram .....	4-153
Figure 4.8-6. CCS Middleware Context Diagram .....	4-162
Figure 4.8-7. CCS Middleware Architecture Diagram.....	4-163
Figure 4.8-8. E-mail Server Context Diagram.....	4-165
Figure 4.8-9. E-mail Server Architecture Diagram .....	4-166
Figure 4.8-10. Virtual Terminal Context Diagram .....	4-169
Figure 4.8-11. Virtual Terminal Architecture Diagram.....	4-170
Figure 4.8-12. Cryptographic Management Interface Context Diagram.....	4-172
Figure 4.8-13. Cryptographic Management Interface Architecture Diagram .....	4-173
Figure 4.8-14. Domains Hierarchy Diagram .....	4-175
Figure 4.8-15. DNS Domains of the EMD Project Diagram.....	4-175
Figure 4.8-16. ECS Topology Domains Diagram .....	4-176
Figure 4.8-17. Domain Name Server Context Diagram .....	4-176
Figure 4.8-18. Distributed Computing Configuration Item (DCCI) CSCI Context Diagram	4-182
Figure 4.9-1. System Management Subsystem Context Diagram .....	4-194
Figure 4.9-2. Network and Enterprise Management Framework Context Diagram .....	4-204
Figure 4.9-3. Network and Enterprise Management Framework Architecture Diagram .....	4-205
Figure 4.9-4. Security Service Context Diagram.....	4-210
Figure 4.9-5. Security Service Architecture Diagram .....	4-211
Figure 4.9-6. Accountability Management Service Context Diagram .....	4-215
Figure 4.9-7. Accountability Management Service Architecture Diagram .....	4-220
Figure 4.9-8. Trouble Ticketing Context Diagram .....	4-237
Figure 4.9-9. Trouble Ticketing Architecture Diagram.....	4-239
Figure 4.9-10. Networker Backup/Restore Architecture Diagram .....	4-245
Figure 4.9-11. Management Software CSCI (MCI) Context Diagram.....	4-247
Figure 4.9-12. Management Software CSCI (MCI) Architecture Diagram .....	4-250

Figure 4.9-13. Inventory/Logistics/Maintenance (ILM) Manager Architecture Diagram .....	4-258
Figure 4.9-14. Software License Manager Architecture Diagram.....	4-262
Figure 4.10-1. DAAC Networks: Generic Architecture Diagram .....	4-281
Figure 4.10-2. SMC Network Architecture Diagram .....	4-282
Figure 4.13-1. Spatial Subscription Server Context Diagram .....	4-302
Figure 4.13-2. Spatial Subscription Server Architecture Diagram.....	4-304
Figure 4.14-1. Data Pool Subsystem Context Diagram.....	4-312
Figure 4.14-2. Data Pool Hardware Context .....	4-315
Figure 4.14-3. Data Pool Insert CSCI Architecture Diagram – Registration .....	4-316
Figure 4.14-4. Data Pool Insert CSCI Architecture Diagram - Publication .....	4-316
Figure 4.14-5. WebAccess CSCI Architecture Diagram .....	4-323
Figure 4.15-1. BMGT Subsystem Context Diagram .....	4-336
Figure 4.15-2. BMGT Architecture Diagram .....	4-339
Figure 4.15-3. BMGT CSCI Context Diagram.....	4-343
Figure 4.15-4. BulkURL CSCI Context Diagram .....	4-350
Figure 4.16-1. OEA Context Diagram.....	4-366
Figure 4.16-2. OEA Architecture .....	4-368

## **List of Tables**

Table 4-1. Memory Management Table .....	4-6
Table 4.1-1. Data Server Subsystem Interface Events.....	4-22
Table 4.1-2. SDSRV CSCI Interface Events .....	4-31
Table 4.1-3. SDSRV CSCI Processes.....	4-36
Table 4.1-4. SDSRV CSCI Process Interface Events .....	4-37
Table 4.1-5. SDSRV CSCI Data Stores.....	4-49
Table 4.2-1. DPL Ingest Subsystem Interface Events .....	4-52
Table 4.2-2. DPL Ingest CSCI Interface Events.....	4-54
Table 4.2-3. DPL Ingest CSCI Processes .....	4-58

Table 4.2-4. DPL Ingest CSCI Process Interface Events .....	4-59
Table 4.2-5. DPL Ingest CSCI Data Stores .....	4-62
Table 4.3-1. Client Subsystem Interface Events.....	4-64
Table 4.3-2. DESKT CSCI Interface Events .....	4-67
Table 4.3-3. DESKT CSCI Processes.....	4-69
Table 4.3-4. DESKT CSCI Process Interface Events .....	4-69
Table 4.3-5. SSI&T Tool Events .....	4-72
Table 4.4-1. Data Management Subsystem Interface Events .....	4-76
Table 4.4-2. Data Dictionary CSCI Interface Events .....	4-83
Table 4.4-3. Data Dictionary CSCI Processes.....	4-85
Table 4.4-4. Data Dictionary CSCI Process Interface Events .....	4-86
Table 4.4-5. Data Dictionary CSCI Data Stores.....	4-91
Table 4.4-6. V0 GTWAY CSCI Interface Events .....	4-94
Table 4.4-7. V0 GTWAY CSCI Processes .....	4-100
Table 4.4-8. V0 GTWAY CSCI Process Interface Events .....	4-101
Table 4.4-9. V0 GTWAY CSCI Data Store .....	4-110
Table 4.5-1. Order Manager Subsystem Interface Events .....	4-112
Table 4.5-2. Order Manager Server CSCI Interface Events .....	4-115
Table 4.5-3. OMSRV CSCI Process.....	4-118
Table 4.5-4. Order Manager Server CSCI Process Interface Events.....	4-118
Table 4.5-5. CSCI Data Stores.....	4-122
Table 4.5-6. Production Module CSCI Interface Events .....	4-123
Table 4.5-7. Production Module CSCI Process .....	4-126
Table 4.5-8. Production Module CSCI Interface Events .....	4-127
Table 4.8-1. Communications Subsystem (CSS) Interface Events.....	4-133
Table 4.8-2. Configuration Registry Server Interface Events .....	4-143
Table 4.8-3. Configuration Registry Server Processes .....	4-145
Table 4.8-4. Configuration Registry Server Process Interface Events .....	4-146

Table 4.8-5. Configuration Registry Server Data Stores .....	4-148
Table 4.8-6. Machine-to-Machine Gateway Server Interface Event .....	4-151
Table 4.8-7. Machine-to-Machine Gateway Server Processes .....	4-155
Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events .....	4-156
Table 4.8-9. CCS Middleware Interface Events .....	4-162
Table 4.8-10. CCS Middleware Processes.....	4-163
Table 4.8-11. CCS Middleware Process Interface Events.....	4-164
Table 4.8-12. CCS Middleware Data Stores.....	4-165
Table 4.8-13. E-mail Server Interface Events.....	4-166
Table 4.8-14. E-mail Server Processes .....	4-167
Table 4.8-15. E-mail Server Process Interface Events .....	4-167
Table 4.8-16. Virtual Terminal Interface Events .....	4-169
Table 4.8-17. Virtual Terminal Processes .....	4-170
Table 4.8-18. Virtual Terminal Process Interface Events.....	4-171
Table 4.8-19. Cryptographic Management Interface Events.....	4-172
Table 4.8-20. Cryptographic Management Interface Processes .....	4-173
Table 4.8-21. Cryptographic Management Interface Process Interface Events .....	4-174
Table 4.8-22. Cryptographic Management Interface Data Stores .....	4-174
Table 4.8-23. Domain Name Server Process .....	4-177
Table 4.8-24. Domain Name Server Process Interface Events .....	4-177
Table 4.8-25. Domain Name Server Data Stores.....	4-178
Table 4.8-26. Infrastructure Libraries.....	4-178
Table 4.8-27. Infrastructure Libraries Group Interfaces.....	4-180
Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events ...	4-186
Table 4.8-29. CSMS CSCI to CSS CSC Mappings.....	4-191
Table 4.9-1. System Management Subsystem Interface Events .....	4-196
Table 4.9-2. Network and Enterprise Management Framework Interface Events .....	4-204
Table 4.9-3. Network and Enterprise Management Framework Processes .....	4-206

Table 4.9-4. Network and Enterprise Management Framework Process Interface Events ....	4-206
Table 4.9-5. Network and Enterprise Management Framework Data Stores.....	4-210
Table 4.9-6. Security Service Interface Events.....	4-211
Table 4.9-7. Security Service Processes .....	4-212
Table 4.9-8. Security Service Process Interface Events .....	4-213
Table 4.9-9. Security Service Data Stores .....	4-214
Table 4.9-10. Accountability Management Service Interface Events .....	4-217
Table 4.9-11. Accountability Management Service Processes.....	4-223
Table 4.9-12. Accountability Management Service Process Interface Events .....	4-224
Table 4.9-13. Accountability Management Service Data Stores.....	4-236
Table 4.9-14. Trouble Ticketing Interface Events .....	4-238
Table 4.9-15. Trouble Ticketing Processes .....	4-240
Table 4.9-16. Trouble Ticketing Process Interface Events.....	4-241
Table 4.9-17. Trouble Ticketing Data Stores .....	4-244
Table 4.9-18. Networker Backup/Restore Processes.....	4-245
Table 4.9-19. Networker Backup/Restore Process Interface Events .....	4-246
Table 4.9-20. Networker Backup/Restore Data Stores.....	4-247
Table 4.9-21. Management Software CSCI (MCI) Interface Events .....	4-248
Table 4.9-22. Management Software CSCI (MCI) Processes .....	4-252
Table 4.9-23. Management Software CSCI (MCI) Process Interface Events .....	4-253
Table 4.9-24. Inventory/Logistics/Maintenance (ILM) Manager Processes .....	4-259
Table 4.9-25. Inventory/Logistics/Maintenance (ILM) Manager Data Stores .....	4-259
Table 4.9-26. Software Change Manager Processes.....	4-260
Table 4.9-27. Software Change Manager Data Stores.....	4-261
Table 4.9-28. Software License Manager Processes .....	4-263
Table 4.9-29. Software License Manager Process Interface Events.....	4-264
Table 4.9-30. Software License Manager Data Stores .....	4-265
Table 4.9-31. ECS Assistant Script Functions.....	4-266

Table 4.9-32. Applications File Format.....	4-269
Table 4.9-33. Cfgparms File Format.....	4-269
Table 4.9-34. Cfgpatch file format .....	4-270
Table 4.9-35. Components file format.....	4-271
Table 4.9-36. Executables file format.....	4-273
Table 4.9-37. Hostmap file format.....	4-273
Table 4.9-38. Installdtypes file format.....	4-274
Table 4.9-39. Packages file format .....	4-275
Table 4.9-40. Sitehostmap file format .....	4-275
Table 4.9-41. Sitemap file format .....	4-276
Table 4.9-42. Subsystems file format .....	4-276
Table 4.9-43. Functions .....	4-277
Table 4.9-44. Start Scripts .....	4-277
Table 4.9-45. Mkcfg Scripts .....	4-278
Table 4.10-1. Internetworking Subsystem Baseline Documentation List .....	4-283
Table 4.10-2. Networking Hardware for EMD Networks .....	4-284
Table 4.11-1. Request Responses .....	4-292
Table 4.11-2. Fault Handling Policies .....	4-293
Table 4.11-3. Server Response versus Restart Temperature .....	4-296
Table 4.11-4. Server Response for Request Re-submission .....	4-297
Table 4.11-5. Server Responses to Client Failures .....	4-298
Table 4.11-6. Client Restart Notification Exceptions.....	4-298
Table 4.11-7. Server Responses to Client Notification .....	4-299
Table 4.13-1. Subscription Server Interface Events .....	4-303
Table 4.13-2. Spatial Subscription Server Processes.....	4-305
Table 4.13-3. Spatial Subscription Server Process Interface Events .....	4-306
Table 4.13-4. Spatial Subscription Server Data Stores.....	4-308
Table 4.14-1. Data Pool Subsystem Interface Events.....	4-312

Table 4.14-2. Use Cases for Data Pool Insert.....	4-317
Table 4.14-3. Data Pool Insert CSCI Process Description .....	4-317
Table 4.14-4. Data Pool ECS Insert CSCI Process Interface Events .....	4-318
Table 4.14-5. WebAccess CSCI Process Description .....	4-324
Table 4.14-6. WebAccess CSCI Process Interface Events .....	4-325
Table 4.14-7. Data Pool Data Stores.....	4-334
Table 4.15-1. BMGT Subsystem Interface Events .....	4-337
Table 4.15-2. BMGT Processes .....	4-340
Table 4.15-3. BMGT Subsystem Interface Events .....	4-344
Table 4.15-4. BMGT CSCI Process Interface Events .....	4-346
Table 4.15-5. CSCI Data Stores.....	4-349
Table 4.15-6. BMGT Subsystem Interface Events .....	4-350
Table 4.15-7. BulkURL CSCI Process Interface Events .....	4-352
Table 4.15-8. CSCI Data Stores.....	4-353
Table 4.16-1. OEA Interface Events.....	4-366
Table 4.16-2. OEA Processes Interface Events .....	4-368

## **Abbreviations and Acronyms**

This page intentionally left blank.

# **1. Introduction**

---

## **1.1 Purpose and Scope**

The purpose of the Segment/Design Specification for the Earth Observing System (EOS) Data and Information System (EOSDIS) Maintenance and Development (EMD) is to provide an overview of the hardware and software subsystems of the project. This document describes the high-level design of each ECS software subsystem implemented to satisfy the allocated and derived functional and performance requirements. This document also provides basic descriptions of the Commercial Off The Shelf (COTS) hardware and software used in the ECS. This document contains :

- Functional overviews of each Computer Software Configuration Item (CSCI)
- Context diagrams of each CSCI
- Interface event descriptions based on the context diagrams
- Process architecture diagrams
- Interface event description tables based on the process architecture diagrams
- CSCI data stores (databases as they relate to the process architecture diagrams)
- CSCI functions allocated to processes. For data servers, this includes descriptions of the functionality offered to clients via the server interfaces. For Graphical User Interface (GUI) applications, it describes the functionality provided to the GUI users
- Specific limitations of the capabilities provided
- Summary of object classes listed by CSCI
- Summary of class libraries listed by CSCI
- Abbreviations and Acronyms

## **1.2 Document Organization**

The remainder of this document is organized as follows:

- Section 2: Related Documentation
- Section 3: System Description
- Section 4: Subsystem Description
- Section 5: Limitations of Current Implementation
- Abbreviations and Acronyms

This page intentionally left blank.

## 2. Related Documentation

---

### 2.1 Parent Documents

The parent documents are the documents from which the scope and content of this Design Specification are derived. These documents are listed below.

423-46-01	EMD F&PRS
423-46-03	EMD Task 101 Statement of Work
194-207-SEI	System Design Specification for the ECS Project

### 2.2 Applicable Documents

Refer to the 900 Series documentation found on the EMD Baseline Information System (EBIS) website: <http://cmdm.hitc.com/baseline/>.

#### 2.2.1 Other Related Documents and Documentation

205-CD-004	Science User's Guide and Operations Procedures Handbook (Release B.0) for the ECS Project
311-EMD-100	Release 7.20 Ingest Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-101	Release 7.20 Science Data Server Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-102	Release 7.20 Storage Management Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-103	Release 7.20 System Management Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-104	Release 7.20 Order Manager Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-105	Release 7.20 Spacial Subscription Server Subsystem Database Design and Schema Specifications for the EMD Project
311-EMD-106	Release 7.20 DataPool Subsystem Database Design and Schema Specifications for the EMD Project
333-EMD-001	Release 7 SDP Toolkit User's Guide for the EMD Project
611-EMD-100	Release 7.20 Mission Operations Procedures for the EMD Project, Section 3.2

625-EMD-101	Release 7.20 Training Material Volume 1: Course Outlines
625-EMD-103	Release 7.20 Training Material Volume 3: Problem Management
625-EMD-108	Release 7.20 Training Material Volume 8: Ingest
625-EMD-109	Release 7.20 Training Material Volume 9: Data Distribution
625-EMD-111	Release 7.20 Training Material Volume 11: Database Administration
193-801-SD4	PGS Toolkit Requirements Specification for the ECS Project (a.k.a. GSFC 423-16-02)
193-WP-118	Algorithm Integration and Test Issues, White Paper for the ECS Project
194-WP-925	Science Software Integration and Test White Paper for the ECS Project
423-16-01	Data Production Software and Science Computing Facility (SCF) Standards and Guidelines
423-42-06	Interface Control Definition for the EOS Data Gateway (EDG): Messages and Development Data Dictionary V0 and ASTER/ECS Message Passing Protocol Specification
RFC 793	Transmission Control Protocol
RFC 768	User Datagram Protocol
RFC 791	Internet Protocol
RFC 1597	Address Allocation for Private Internet
	WWW page is <a href="http://cmdm.east.hitc.com">http://cmdm.east.hitc.com</a>

## 2.3 Information Documents Not Referenced

The documents listed below, while not directly applicable, help in the maintenance of the delivered software.

423-41-02	Goddard Space Flight Center, Functional and Performance Requirements Specification for the Earth Observing System Data and Information System Core System
423-41-03	EOSDIS Core System Contract Data Requirements Document
230-TP-002	Interface Control Document Between the ECS and the Product Distribution System Information Server (PDSIS)
505-41-21	IRD between ECS and NASA Institutional Support System (NISS)
505-41-30	ICD between ECS and Version 0 System for Interoperability

505-41-34	ICD between ECS and ASTER Ground Data System
505-41-38	ICD between ECS and EOS-AM Project for AM-1 Spacecraft Analysis System
505-41-40	ICD between ECS and the GES DAAC
423-41-57-1	ICD between ECS and SIPS, Volume 1: ECS ACRIM III SIPS Data Flows
423-41-57-2	ICD between ECS and SIPS, Volume 2 SAGE III SCF Data Flows
423-41-57-3	ICD between ECS and SIPS, Volume 3 ASTER OSF
423-41-57-4	ICD between ECS and SIPS, Volume 4 ASTER DEM
423-41-57-5	ICD between ECS and SIPS, Volume 5 MOPITT Data Flows
423-41-57-6	ICD between ECS and SIPS, Volume 6 MODIS (MODAPS)
423-41-57-7	ICD between ECS and SIPS, Volume 7 AMSR-E
423-41-57-9	ICD between ECS and SIPS, Volume 9 MTMGW
423-41-57-10	ICD between ECS and SIPS, Volume 10, TES Data Flows
423-41-57-11	ICD between ECS and SIPS, Volume 11, ICESat Data Flows
423-41-58	ICD between the ECS and LP DAAC
423-ICD-EDOS/EGS	ICD between EDOS and EGS
423-EDOS-0211.0001R2	Interface Requirements Document between EDOS and EGS Elements

## 2.4 ECS Tool Descriptions

### 2.4.1 Rational Rose

The Rational Rose tool provides support for object-oriented analysis and design. In particular, the Rose tool provides support for controlled-iterative or component-based development. The Rose tool is used on the EMD Project to document the object-oriented elements of the design using class diagrams, use-case diagrams, interaction diagrams, component diagrams, and object diagrams. The Unified Modeling Language (UML) is the methodology used on the EMD Project for all design activities (although the Rose tool also supports the Booch '93 Methodology or the Object Modeling Technique (OMT) as well).

The Rose tool can also be used to reverse engineer code developed that lacks supporting documentation to get as-built object diagrams.

Before using the Rational Rose tool, see “Rational Rose 98, Using Rose” for important tool usage and reference information. In addition, the following references can be obtained and used:

- (1) "Unified Method for Object-Oriented Development," by Grady Booch and Jim Rumbaugh (version 1.1, Rational Software Corporation) for an introduction to the respective method's notation, semantics, and process for object-oriented analysis and design.
- (2) the second edition of "Object-Oriented Analysis and Design with Applications" by Grady Booch, (Benjamin/Cummings, 1994)
- (3) "Object-Oriented Modeling and Design" by James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy and William Lorensen, (Prentice-Hall, 1991)
- (4) "UML Distilled: Applying the Standard Object Modeling Language" by Martin Fowler with Kendall Scott, Foreword by Grady Booch, Ivar Jacobson, and James Rumbaugh (Addison Wesley Longman, Inc., 1997)

#### **2.4.2 ClearCase Baseline Manager Configuration Management Tool**

ClearCase Baseline Manager (CBLM) consists of the ECS baseline data and a Graphical User Interface (GUI) used to control the ECS baseline.

The data comes from two sources:

- 1) Existing Release Notes (914-TDA-xxx) Machines Impacted data
- 2) Newly approved CCRs

Control Item Identifiers (CIDs) consist of an 8-digit integer with a "b" prefix (e.g., b00083456). Each COTS S/W product has its own CID. Because CIDs are mapped to ECS hosts, it was decided to represent information within ClearCase as elements. For the ClearCase CID elements, the comma separated variable (CSV) format was chosen, as this format is easily ported into and from other COTS S/W products, specifically Microsoft Access and Excel.

The ClearCase configuration specification chosen was the simplest, or the default configuration specification. A view, CM\_MASTER, was created with the default configuration specification to manage the data records. The CID records (checked in ClearCase elements) are located in the /ecs/cm VOB at /ecs/cm/CIDs. This directory currently contains the 256 records that correlate to XRP-II's COTS S/W CIDs.

Another important data construct within CBLM is the notion of the Machines Impacted file, and a CCR identified construct, which maps CIDs to hosts. Each Configuration Change Request (CCR) affecting the baseline contains information about 1 or more CIDs. Also, the CCR contains information regarding the hosts receiving the COTS S/W (CID). So the CCR has a construct that in its simplest form is one "CID\_MAP" file, and one Machines Impacted (MI) file. The "CID\_MAP" file is a simple lookup table. In this case, there is only one entry. The entry contains first a valid CID, followed by one or more blanks, then the name of the "MI" file. In this case, the MI filename is "MI." The MI file, contains a list of valid ECS hosts having the COTS S/W identified within the CID. So a CCR (07-1234) to place a COTS S/W (e.g., Acrobat Reader), onto host e4eil01, would have an MI file containing one host, e4eil01, and one CID\_MAP. If the Acrobat Reader software is CID b00081234, the CID\_MAP file would contain:

“b00081234 MI”

And the MI file contains:

“e4eil01”

The CCR would be found at:

/ecs/cm/CM/2007CCRs/1234/, a directory

Under this directory is found the two files, “CID\_MAP” and “MI.” Note that there is always only one CID\_MAP file for each CCR, but that the CID\_MAP may contain more than one entry. The simplest example of this is when a COTS S/W product needs to be mapped to SGI, Sun, and Linux hosts. There would be three MI files, “MI\_SGIs”, “MI\_Suns”, and “MI\_Linux” for example. The CID\_MAP would contain three entries, one mapping the SGI hosts to the SGI CID, one mapping the Sun hosts to the Sun CID, and one mapping the Linux hosts to the Linux CID.

As approved CCRs are required to change the CBLM data state, the effectivity date is then defined as the CCR approval date. This is the date the change becomes valid. The next construct, named the “Sequencer”, is the table providing the history of change. The last approved CCR is at the end of the table. As new CCRs get approved, they simply get concatenated to the end of the list in time order. The Sequencer is an executable script.

The last construct is the “dartboard.” Conceptually, the “dartboard” is a directory within ClearCase, at /ecs/cm/BLM/dartboard/. All ECS hosts are listed as files in the /dartboard/ directory. In conclusion, then data constructs are:

CIDs

CCR directories

CID\_MAPs and MIs under the CCR directories

Sequencer

Dartboard

The way these pieces all work together is now briefly discussed.

When a CCR is approved that affects the baseline, a CCR is checked out. The /ecs/cm/CIDs/ ClearCase directory is checked out. The new CID is created and populated with the information present on the CCR form. The new CID number then has a ClearCase element created, and the first version becomes this new CID. The /ecs/cm/CIDs/ directory is then checked back in. Next, the MI file must be prepared. Within the CCR directory, two new files are “made” (cleartool mkelem –eltype text\_file –nc CID\_MAP MI). The hosts, which are to get the COTS S/W, are entered into the MI file, then the file is checked in as the first version. Next, the CID\_MAP file is created, mapping the new CID number to the MI file. The CID\_MAP file is checked in, then the CCR file is checked back in. This work gets the CCR information locked into ClearCase.

Next, the Sequencer file is edited to show the new CCR number at the end. This action allows the CCR’s MI and CID\_MAP files to overlay onto the ClearCase baseline. This is accomplished

by echoing the contents of the CID (in file /ecs/cm/CIDs/b00083123) onto each of the hosts specified with the /ecs/cm/CM/2007CCRs/1234/MI file. This data is written to the hosts files with the dartboard, located at /ecs/cm/BLM/dartboard.

Once the data has been applied to the dartboard, subsequent scripts then produce the output reports. In conjunction with the current hosts list, the scripts obtain all of the valid hosts of the site, and basically reformat the data within the dartboard files into reports. Information is added to the reports, including the CCR number, related Release Notes documentation, and the CCR approval date.

The reports are written to the /ecs/cm/BLM/reports directory. Subsequent “expect” scripts then scp those reports to specific locations on the EBIS server, c4cbl02, and then the reports are replicated to each of the 5 remote sites (SMC and 4 DAACs).

The languages used in this tool are “sh”, “csh”, “expect”, and C. Also, “.grp” files are used to represent the ClearCase GUIs. These files are text files that are dynamically generated at the time that the GUI is launched. Code has been reused from two sources, the DeliveryTool, which is used to prepare and send data to the sites, and the replication scripts, which are used to replicate data from the Landover EBIS server c4cbl03, to the protected (SMC, DAACs) servers.

## **3. System Description**

---

### **3.1 Mission and Release 7.20 Objectives**

The Mission of the National Aeronautics and Space Administration's Earth Science Enterprise is to develop a scientific understanding of the total Earth System and its response to natural or human-induced changes to the global environment to enable improved prediction capability for climate, weather and natural hazards. The vantage point of space provides information about Earth's land, atmosphere, ice, oceans and biota that is obtained in no other way. Programs of the enterprise study the interactions among these components to advance the new discipline of Earth System Science, with a near-term emphasis on global climate change. The research results contribute to the development of sound environmental policy and economic investment decisions.

The Earth Observing System Data and Information System (EOSDIS) Core System (ECS) has been designated as the ground system to collect, archive, produce higher-level data products and distribute data for the Earth System Science mission.

#### **3.1.1 Release 7.20 Capabilities**

The ECS capabilities have been developed in increments called formal releases. Release 7.20, which is managed by Configuration Management, is a formal release. It is a collection of new and updated capabilities provided to the users of the system and is described here to show the progress of system enhancements. The ECS collects and stores, processes, archives and distributes scientific data from six different platforms (satellites). In the following sub-sections, the platforms and instruments from which scientific data is collected are identified, the type of data ingested and archived is presented, search and order capabilities for scientific data, how data is distributed and processed, system architecture and operation, system security and Distributed Active Archive Center (DAAC) and external system support are described. Other capabilities provided by Release 7.20 include processing the data obtained, distributing raw or processed data as requested, quality assurance of processed data, supporting communication networks, and systems monitoring via interfaces with the ECS operations staff.

Release 7.20 unique capabilities and modifications include:

- Data Pool Insert Enhancements – In Release 7.20, collisions and replacements will no longer occur when granules are inserted into the non-public Data Pool; rather, granule files will be renamed by appending a suffix during insert if a potential collision is detected. With this release, the Data Pool Action Driver will dispatch these activities differently. A number of the above steps, namely copying granule files, checksumming and band information extraction, will be removed from the DPIU and dispatched as separate operations. The DPAD will use separate hosts, called ECS Service Hosts, for the archive copy and checksumming operations to off-load work from the Data Pool platform

(band extraction is not a high workload and will be performed on the Data Pool platform itself.)

- SIPS Ingest Into Data Pool – This capability provides a new ingest and archiving service for the Data Pool and is defined in several related tickets. This covers the various variants of the SIPS ingest interface. The SIPS interface is also used to ingest S4P outputs and to support cross-DAAC ingest. This covers all SIPS ingest including that requiring secure transfers. It supports the ingest protocol known as ‘Polling with Delivery Record’.
- Ingest of Level 0 Data from EDOS into the Data Pool – This capability includes the ability to perform metadata extraction as required for ingest of L0 data from EDOS as part of the general preprocessing step required for Data Pool Ingest.

The ticket also adds the ability to accommodate the variations in ingest protocol and file formats specified in the EDOS-EGS ICD, ESDIS document 423-ICD-EDOS/EGS..

- Ingest of ASTER L1A and Browse into Data Pool – Release 7.20 adds the ability to perform metadata extraction as part of the general preprocessing step for the Data Pool Ingest capability. The capability complies with the ESDIS CCR for incorporation into ESDIS Document 423-41-58, ICD between ECS and the LP DAAC. The ICD Between ECS and ASTER Ground Data System, ESDIS Document 505-41-34 is obsolete.
- Physical Media Production with Luminex – In release 7.20, the existing Rimage units used for physical media distribution are replaced by Luminex CDROM/DVD production units, which will run on Linux hosts.
- Distribution of integrated browse products requested via the Version 0 Gateway – Starting with Release 7.20, staging browse files for integrated browse requests will no longer involve the SDSRV. Rather, the V0 Gateway will stage the browse file (for ‘order only purposes’) via the Data Pool and then return it to the EDG.
- Distribution Of Metadata in Either XML or .met Format – Release 7.20 adds the capability to OMS to provide metadata files in either format, depending upon configuration settings by the DAAC.
- OMS Distribution of Data via scp – Distribution notices (DNs) for successful scp transfers will continue to be distributed to the user using secure copy. However, DNs for failed scp transfers, as well as DNs which fail to transfer via secure copy, will result in the email distribution of the failed DN to the email address associated with the request. This represents a change from the existing behavior, which distributed the failed DN to an email address configured as part of the DDIST subsystem.

Note that requests for scp distribution are currently being submitted only by the Spatial Subscription Server (SSS) for certain subscriptions designated for scp-distribution by the DAAC. The availability of secure copy as a media type is not expected to expand beyond its current scope.

- OMS Distribution of Externally subsetted Products Requested via V0GWY – In Release 7.20, the V0 Gateway will partition the orders like today, but register each subsetting request with the OMS before sending it to the external subsetter. The product will then be distributed via the OMS.
- Web Accessible Order Status and History – The Order Status Interface is a web-accessible interface that allows end-users and ECHO users to check on the status of in-progress or completed orders. The user can obtain the status of a single order or request the history of all orders submitted over a specified date range.
- Add spatial subsetting support for AMSR-E products – Two AMSR-E GRID data sets (Polar Stereographic (PS), and Lambert Azimuthal (LA)) are in two different input projections from that of the MODIS SIN projection which is the bulk of the traditional support. "resample", which handles GRID data, has to modified to handle spatial subsetting for these two projections. A previous delivery to implement these two AMSR-E GRID data sets handled only the full image case. Also, pixel size change (from default) is handled for these two projections.
- EPD Must handle multiple ESDTs in output products PDR from subsetters – This capability, originally provided as a TE to Release 7.11, is being formally delivered in Release 7.20. This provides support for multi-esdt / multi-granule requests from external processors.
- Added BMGT command line capability to input MISR Browse dbId – This capability, originally delivered as a TE to Release 7.11, allows for the specification of files to be processed by specifying the db id.
- DP Maintenance Script - Remove Invalid collections – A new script, EcDIRemoveCollection.pl, was provided as a TE to Release 7.11 and is being formally delivered in Release 7.20. This allows for the removal of collections from the DPL via a command line script.
- BMGT should export GCMD DIF\_ID in collection metadata to ECHO – Modified the dtd and schema from input provided by ECHO personnel.

### **3.1.1.1 ECS Support of Instruments by Platform**

- The Meteor 3 platform supports the Stratospheric Aerosols and Gas Experiment III (SAGE III) instrument
- The ACRIMSAT platform supports the ACRIM III experiment
- The Terra (AM-1) platform supports the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), Multi-Angle Imaging SpectroRadiometer (MISR), Moderate Resolution Imaging SpectroRadiometer (MODIS) and Measurements of Pollution in the Troposphere (MOPITT) instruments

- The Aqua (PM-1) platform supports the Moderate Resolution Imaging SpectroRadiometer (MODIS) and Advanced Microwave Scanning Radiometer (AMSR) instruments
- The Ice, Cloud and Land Elevation satellite (ICESat) platform supports the Geoscience Laser Altimeter System (GLAS) instrument
- The AURA platform supports the Tropospheric Emission Spectrometer (TES) instrument

### **3.1.1.2 Ingest and Archive Capabilities**

The following data is ingested and archived in the ECS from the various instruments described in section 3.1.1.1:

- Ingest of science and engineering data from the EOS Data and Operations System (EDOS)
- Ingest of Product Generation Executable (PGE) software from Science Computing Facilities (SCFs) either electronically or via media tape
- Ingest of ASTER Level 1A/1B data
- Ingest of FDS (formerly FDD) orbit data
- Ingest of SAGE III MOC Level 0 data
- Ingest of SAGE III SCF higher-level products via the SIPS interface
- Ingest of Data Assimilation System (DAS) HDF-EOS data via standard polling with DR
- Ingest of MODIS higher-level products via the SIPS Interface
- Ingest of MOPITT SCF Level 0 data via the SIPS interface
- Ingest of SDPS resident data across a mode in the same DAAC or across DAACs
- Ingest of ACRIM Level 0 and higher-level data from the ACRIM SCF via the SIPS interface
- Ingest of higher-level AMSR data products from the AMSR SCF
- Archive of ICESat GLAS Level 1, Level 2 and Level 3 and ancillary data at the NSIDC DAAC
- Archive of TES Level 1, Level 2 and Level 3 data
- Archive of products previously processed and archived

### **3.1.1.3 Search and Order Capabilities**

The ECS provides the following capabilities for search and ordering of data from the archive:

- Machine-to-Machine Gateway support between SIPS and ECS for data orders
- Directory and inventory search, including a user browse capability via the Version Zero (V0) System user interface
- Provide access to non-science data collections by a limited number of attributes and values

- V0 Gateway support for client requests for searches based on full ECS core metadata and product specific attributes and data subsetting for a limited number of data types
- Tracking order processing status via the System Management Subsystem (MSS)
- Configurable parameters to control the number of granules returned from a single search request
- Handling of variations on search areas and product-specific spatial representations
- Managing of orders via the V0 Gateway from the EOS Data Gateway (EDG), ECS Clearing House (ECHO), and the orders submitted by the Spatial Subscription Server (SSS)
- The SSS provides an operator the interface to place standing orders (subscriptions) based on an ECS event and manage subscription status
- The Data Pool provides an operator the interface to manage insert processes, queues, collection groups and collection themes for ECS and non-ECS collections

#### **3.1.1.4 Data Distribution Capabilities**

The ECS provides the following Data Distribution capabilities for users:

- Support writing files to CD-ROMs and Digital Linear Tape drives for distribution
- Support File Transfer Protocol (FTP) Push or Pull Subscriptions for users
- Support distributing science data products via FTP, CD-ROM, DVD, and DLT. (**Note:** physical media may not be available through all ordering applications.)

#### **3.1.1.5 Data Processing Capabilities**

The ECS provides the following capabilities for user/operator data processing options:

- Support the archive of products previously produced and archived
- Provide capability for operator deletion of granules
- Support Quality Assurance (QA) processing of Terra (AM-1) science data products
- Automated support for on-demand requests for ASTER processing
- Provide capability to associate the ASTER browse granule for the L1A product with ASTER L1B products

#### **3.1.1.6 System Operation and Architecture**

The ECS provides the following capabilities to support the system operations and processing architecture used to provide data and services for users:

- Provide capability for operator deletion of granules, their associated metadata and browse files

- Provide the associated communications network interfaces with the SCFs
- Support managing the startup and shutdown of system network components, user registration and profile administration, database and archive administration, system data and file back-up and restores, system performance tuning and resource usage monitoring, and other routine operator duties
- Support the display of browse data as a result of a single user request from the search results screen
- Operations support to update certain ESDT attributes without requiring the deletion of the data collection
- Provide ESDTs to support MODIS, and AMSR on Aqua (PM-1)
- Provide the capability for editing of ECS core attribute values
- Support the consolidation of trouble tickets using TestTrack Pro
- Provide fault recovery for mode management
- Provide the capability for startup and shutdown of an entire mode
- Provide the capability for the deletion of science data from the archive
- Provide the capability for the installation of ESDTs to insert and acquire archived data
- Provide the capability for the persistence of asynchronous acquire requests
- Provide for the storage of event information into the SDSRV database instead of flat files
- Provide the capability for the monitoring of the usage of memory
- Provide COTS packages to allow operations to generate customized reports from ECS databases
- Provide a single configuration registry database to replace the numerous ECS application configuration files
- Provide for the insertion of ECS and non-ECS granules into the Data Pool

### **3.1.1.7 Security**

The ECS provides the following capabilities for system security:

- Encryption of passwords in ECS databases
- User authorization checks to restrict data set access at the granule level based on data quality information
- SDP Toolkit support for thread safe concurrent processing by the science software
- Secure Transfer of data files from Data Providers upon request
- System data and file backups and restores

### **3.1.1.8 DAAC/External System Support**

ECS Release 7.20 will be distributed to three site locations including:

1. The DAAC at the Langley Research Center (LaRC),
2. The Land Processing DAAC (LP DAAC), and
3. The DAAC at the National Snow and Ice Center (NSIDC)

In addition, ECS Release 7.20 requires that ECS Release 7.11 be operational at the System Management Center (SMC), located at the Goddard Space Flight Center (GSFC). Release 7.20 will not be distributed to the SMC.

The ECS Release 7.20 communications network includes the National Aeronautics and Space Administration (NASA) and the NASA Integrated Services Network (NISN). These portions of the network are physically located at the SMC and at the DAAC sites. The communications network connects ECS to data providers at the EDOS, NOAA Affiliated Data Center (ADC), and the EOSDIS Version 0 system.

The data users for Release 7.20 are the science user community connected to the SMC, the three DAACs, the SCFs, and the MODAPS.

1. SMC Support:
  - SMC capabilities include overall ECS system performance monitoring, coordinating, and setting system-wide policies and priorities
2. LaRC Support:
  - ECS Release 7.20 provides a communications network and data/information management support for MISR instrument data including the receipt of MISR level 0 data and the LaRC archive and distribution of levels 1, 2 and 3 data and data products
  - ECS Release 7.20 provides a communications network and data/information management support for MOPITT instrument data including the receipt of MOPITT level 0 data, the LaRC archive, and distribution of levels 1, 2 and 3 data
  - ECS Release 7.20 provides a communications network and data/information management support for TES instrument data including the receipt of TES level 0 data and the LaRC archive, and distribution of levels 1, 2 and 3 data
  - LaRC DAAC capabilities include:
    - Ingest of MISR, TES, and MOPITT Level 0 and related ancillary data
    - Archival, and distribution of the higher-level products for MISR
    - Receipt of higher-level MOPITT products from the MOPITT SCF, via the SIPS interface, for archival and distribution

- Receipt of SAGE III products from the SCF, via the SIPS interface, for archival and distribution
- Receipt of ACRIM products (Level 0 and Level 2 data) from the SCF, via the SIPS interface, for archival and distribution
- Receipt of TES Levels 1-3 data including algorithm and associated software packages, metadata, production histories, ancillary data and Quality Assessment (QA) data for archival and distribution

3. LP DAAC Support:

- ECS Release 7.20 provides a communications network and data/information management support for ASTER instrument data including the receipt of ASTER level 1A data electronically at LP DAAC from Japan, and distribution of higher level ASTER products by LP DAAC
- LP DAAC capabilities include:
  - Ingest of ASTER Level 1A/1B, with ancillary data needed for production
  - Archival and distribution of ASTER products
  - Receipt of higher level MODIS land products from MODAPS, via the SIPS interface, for archival and distribution

4. NSIDC Support:

- AMSR-E instrument data including the receipt of level 0 data from EDOS at ECS, and the NSIDC archive and distribution of levels 1, 2 and 3 data. The Level 1A data is received from the NSIDC V0 DAAC while the level 2 and 3 data is received from the AMSR-E SCF via the SIPS interface
- AMSR-ADEOS II Level 1A data is received from the NSIDC DAAC and archived and distributed using ECS.
- ECS Release 7.20 supports the ingest of ICESat GLAS level 1, level 2, level 3 and ancillary input data for archive and distribution at the NSIDC DAAC using the standard SIPS interface. The ECS also archives GLAS level 0 data received from EDOS
- NSIDC DAAC capabilities include:
  - Receipt of higher-level MODIS snow and ice products from MODAPS, via the SIPS interface, for archival and distribution
  - Ingest of AMSR-E Level 0 data and related ancillary data
  - Receipt of the AMSR-E and AMSR-ADEOS II higher-level products via the SIPS interface, for archival and distribution

- Ingest of GLAS Level 0 data and related ancillary data for archival and distribution.
- Receipt of the GLAS higher level products from the SCF, via the SIPS interface, for archival and distribution

5. SCF Support:

- The MOPITT higher-level products are generated at the SCF and provided to the ECS via the SIPS interface
- ECS Release 7.20 supports receiving SAGE III Level 0 data and higher level products from the SCF via the SIPS interface
- ECS Release 7.20 supports receiving ACRIM L0 data and higher level products from the SCF via the SIPS interface

6. MODAPS Support

- ECS Release 7.20 provides a communications network and data/information management support for MODIS instrument data including: archive and distribution of higher level data from the MODIS Data Processing System (MODAPS)

## **3.2 Release 7.20 Architecture Overview**

The ECS Release 7.20 architecture comprises the logical items listed here. Commercial Off The Shelf (COTS) software and hardware are used, to the extent possible, to implement the ECS functionality of these logical items.

- System
- Segments
- Subsystems
- Computer software configuration items (CSCIs)
- Computer software components (CSCs)
- Processes

ECS Release 7.20 was built of the following two segments.

- CSMS – Communications and Systems Management Segment
- SDPS – Science Data Processing Segment

Each segment was in turn built of the following subsystems:

- CSMS:
  - CSS – Communications Subsystem
  - ISS – Internetworking Subsystem

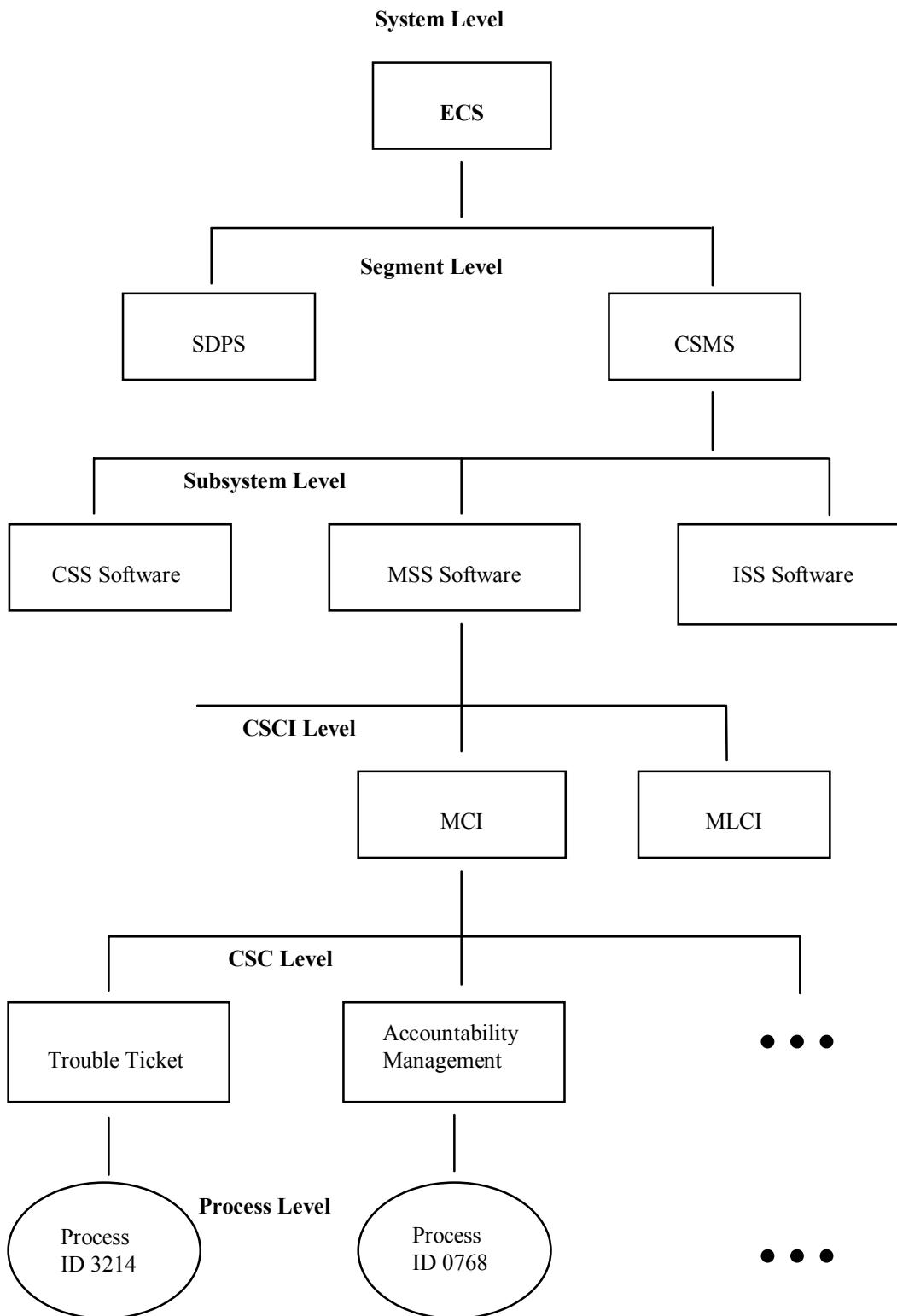
## MSS – System Management Subsystem

- SDPS:
  - BMGT – Bulk Metadata Generation Tool Subsystem
  - CLS – Client Subsystem
  - DMS – Data Management Subsystem
  - DPL – Data Pool Subsystem
  - DPL INGEST – Data Pool Ingest Subsystem
  - DSS – Data Server Subsystem
  - OMS – Order Management Subsystem
  - SSS – Spatial Subscription Server Subsystem

### **Hierarchical Definitions**

System:	A stand-alone composite of hardware, facilities, material, software, services, and personnel required for operation based upon a defined set of system level requirements and designed as a related set of capabilities and procedures.
Segment:	A logical and functional subset of related capabilities, implemented with COTS hardware and COTS and custom developed software to satisfy a defined subset of the system level requirements.
Subsystem:	A logical subset of Segment related capabilities, implemented with COTS hardware and COTS and custom developed software to satisfy a defined subset of segment level requirements.
CSCI:	A logical subset of Subsystem related capabilities, implemented with COTS and custom developed software to satisfy a defined subset of the subsystem level software requirements.
CSC:	A logical subset of CSCI related capabilities, implemented with COTS and custom developed software to satisfy a defined subset of the CSCI level software requirements.
Process:	A logical and functional set of software, written in a specific order and in a defined manageable size to manipulate data as part of a product-generating algorithm. A process is a separately compiled executable (i.e., binary image). A process can use infrastructure library calls, system service calls, COTS service calls, and application programming interfaces to manipulate data to generate products.

Figure 3.2-1 is a hierarchical software diagram. The hierarchical software diagram depicts an example of the decomposition levels used in the ECS design and described in this document. The diagram is also a graphical representation of the terms just described.

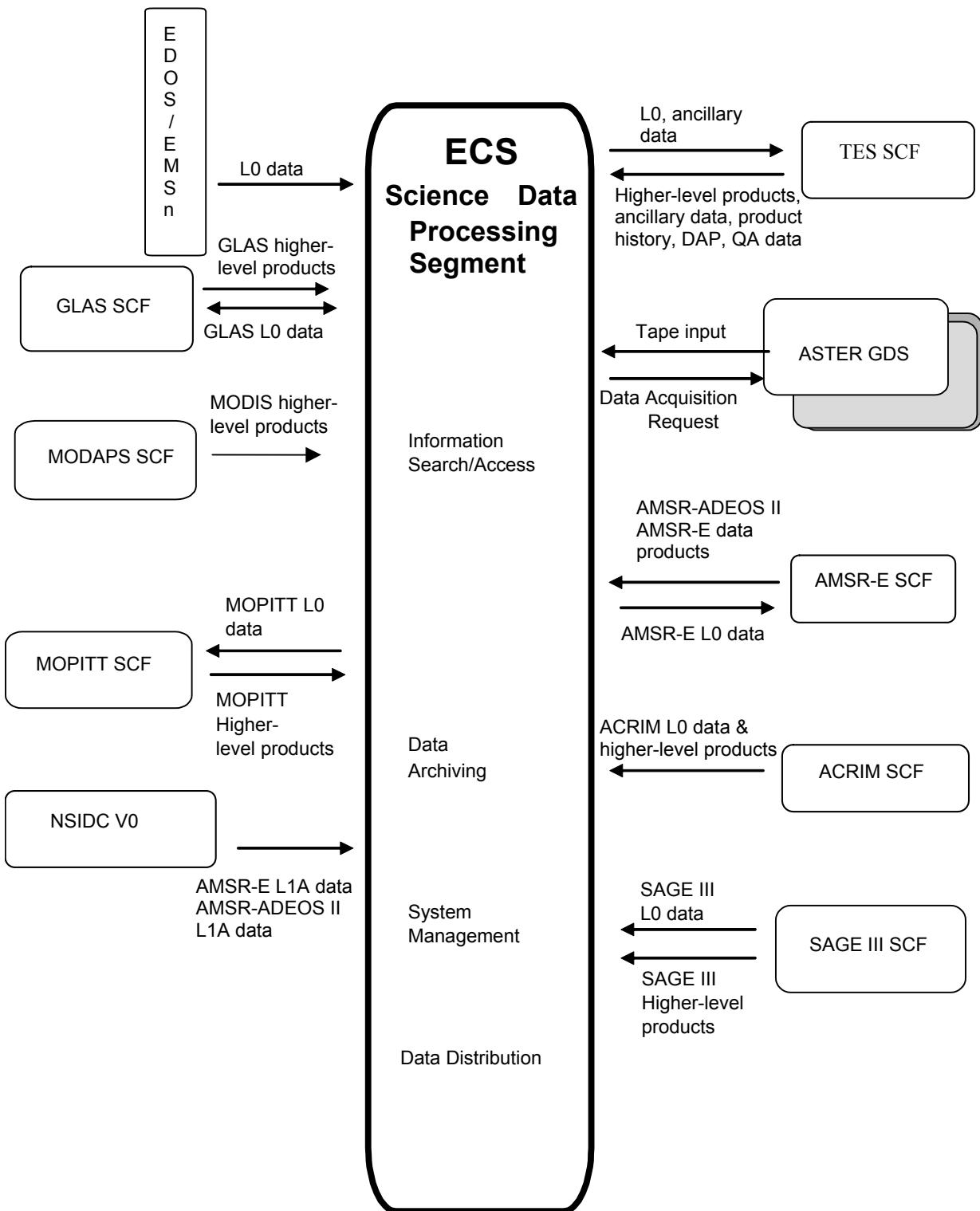


**Figure 3.2-1. Example Hierarchical Software Diagram**

### **3.2.1 Release 7.20 Context Description**

ECS Release 7.20 provides the capability to collect and process satellite science data as depicted in Figure 3.2-2.

The Science Data Processing and Communications and Systems Management are the two segments of Release 7.20 described in this document. The Science Data Processing Segment (SDPS) provides science data ingest, search and access functions, data archive, and system management capabilities. The SDPS receives Terra (AM-1) and Aqua (PM-1) Level 0 science data from EDOS. The SDPS exchanges data with affiliated data centers to obtain science and other data (i.e., engineering and ancillary) required for data production. Science algorithms, provided by the Science Computing Facilities (SCFs), are archived for distribution. The Communications and Systems Management Segment (CSMS) provides the communications infrastructure for the ECS and systems management for all of the ECS hardware and software components. The CSMS provides the interconnection between users and service providers within the ECS, transfer of information between subsystems, CSCIs, CSCs, and processes of the ECS.



**Figure 3.2-2. Release 7.20 Context Diagram**

The remaining sections of this document provide an overview of Release 7.20 design and as such do not deal specifically with the configuration of components at each EOSDIS site. For more information on the site unique configurations, refer to the 920-series of General documents. Each of the segments consists of subsystems as specified in Section 3.2.

### **3.2.2 Release 7.20 Architecture**

#### **3.2.2.1 Subsystem Architecture**

The ECS SDPS subsystems are depicted in Figure 3.2-3. A subsystem consists of the Commercial Off The Shelf (COTS) and/or ECS developed software and the COTS hardware needed for its execution. The SDPS subsystems can be grouped into a ‘Push’ or ‘Pull’ category of functionality with the exception of DSS. As shown in the subsystem architecture diagram, the information search and data retrieval makes up the ‘Pull’ side of the ECS architecture/design and consists of the CLS, DMS, OMS, SSS, DPL and also uses the DSS functionality described on the ‘Push’ side of the ECS architecture. Data capture (ingest of data), storage management, planning and data processing of satellite or previously archived data from other sites make up the ‘Push’ side of the ECS architecture/design and consists of the DSS, DPL, DPL INGEST, and OMS. This document describes the software and hardware components of each subsystem. However, since the hardware configurations differ between the sites, the hardware descriptions in this document are at a generic level. Specific hardware and network configurations for each site are documented in the 920 and 921 series technical documents.

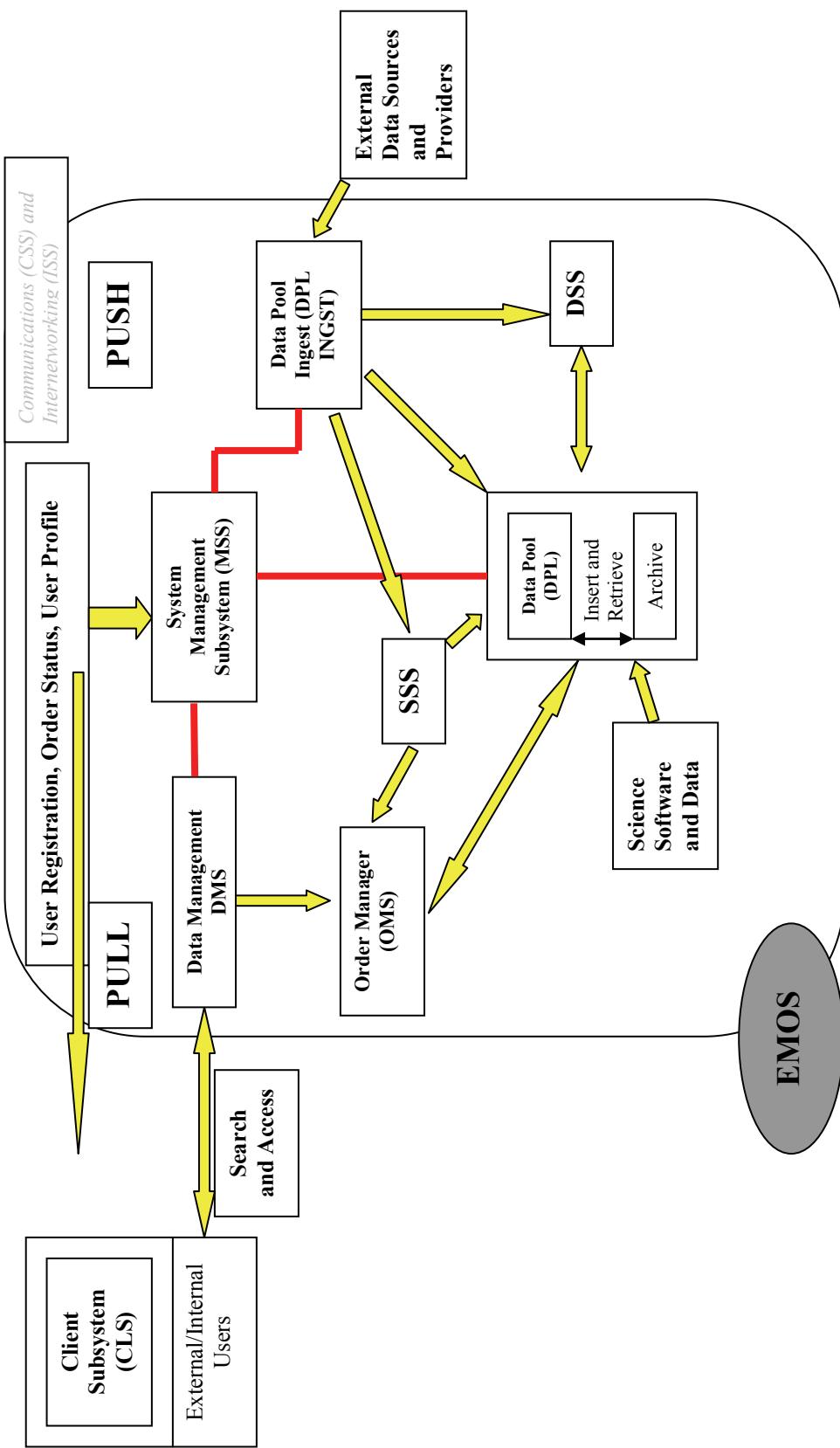


Figure 3.2-3. Subsystem Architecture Diagram

The ECS SDPS architecture/design consists of:

- BMGT exporting inventory status information to ECHO.
- CLS providing user interfaces for data search and retrieval as well as order status to science users and operators.
- DMS providing support for data search and retrieval across all ECS sites. The DMS also provides a gateway as the interface to the Version 0 (V0) Information Management System (IMS) using the V0 IMS protocol.
- DPL supporting the search, order, and distribution of selected granules with associated metadata and browse granules (if available).
- DPL INGEST service will handle the SIPS ingest interface, S4P, cross-DAAC ingest, EDOS ingest, ASTER Ingest and Polling without Delivery Record specifically for EMOS.
- DSS with the functions needed to manage the inventory of archived data.
- OMS managing all orders received from the DMS V0 Gateway (i.e., from EDG and ECHO), the Machine-to-Machine Gateway, and the Spatial Subscription Server.
- SSS supporting the creation, viewing and updating of subscriptions and the creation, viewing and deletion of bundling orders (specification of distribution packages and criteria for package completion).

CSMS – The following subsystems are the CSMS subsystems, which interact with and support the SDPS to complete the ECS architecture.

1. The MSS with:

- Hardware and software baseline and configuration management
- Trouble ticketing and nonconformance report (NCR) tracking
- Fault and performance monitoring for networks, platforms, and software applications
- User account management and user order tracking

2. The CSS with:

- Control Center System (CCS) Middleware provides a common Name Server, which packages the common portions of the communication mechanisms into global objects to be used by all subsystems. The Name Server provides a set of standard CCS Proxy/Server classes, which encapsulates all of the common code for middleware communications (e.g., portals, couplers, RWCollectables, etc.)
- Libraries with common software mechanisms for application error handling, aspects of recovering client/server communications; Universal References to distributed objects and interfaces to e-mail, file transfer and network file copy capabilities

3. The ISS with:

- Networking hardware devices (e.g., routers, switches, hubs, cabling, etc.) and their respective embedded software. For more information on site unique configurations, refer to the 920-series of General documents

This page intentionally left blank.

## 4. Subsystem Description

---

### Design Description Organization and Approach

This section presents a subsystem-by-subsystem overview description of the “as-built” EMD. The current high-level design information is provided for the Hardware Configuration Items (HWCIs), Computer Software Configuration Items (CSCIs), and Computer Software Components (CSC) for each subsystem and is being delivered to the DAACs in drop increments.

The CSMS subsystem descriptions include:

- Subsystem functional overviews with a subsystem context diagram and a table of interface event descriptions
- CSCI descriptions with a context diagram and a table with interface event descriptions
- Architecture Diagrams, Process Descriptions, and Process Interface Event Tables. The Architecture Diagrams show the processes of the CSCI/CSC and how these processes connect with other CSCIs and CSCs of the same subsystem and the interfaces with other subsystems and external entities such as Operations, External Data Providers and Users.
- Data Store descriptions for each CSCI in the CSMS subsystem. The Data Stores are identified with the software name and shown in the architecture diagrams either as single data stores or as a group of data stores with a generic name such as “Data Stores” or “database”
- Hardware descriptions of the subsystem hardware items and the fail-over strategy

The convention used for Context and Architecture diagrams includes using circular shapes to show the subject subsystems, CSCIs, CSCs, or processes (with name in bold), elliptical shapes to show associated CSCIs, CSCs, or processes within a given subsystem and squares or rectangles to show external subsystems, CSCIs, CSCs, and processes. Data stores are shown using the data store or database name with horizontal lines, one above and one below the name. An interface event is data, a message (which includes a notification or status); a command, request or status code passed between subsystems, CSCIs, CSCs, or processes. The convention used to identify events is a straight line between two objects labeled with a phrase beginning with an action-oriented word to best describe the event. The arrow on the event line indicates an origination point and to where the event is directed. A direct response to an event is not always shown on the diagram because sometimes there is no response (e.g., for an insert or delete request) and other times the response comes from another part of the EMD. Interface events are identified in the interface event or process interface tables starting with the interface event at the top or middle of the diagram and going clock-wise around the diagram. The external interface subsystem is identified in the interface event description and is in bold to assist with the location of the interface events on the diagram. If there are two items in bold, there are two different interfaces (Subsystems, CSCIs, or CSCs) requesting the same interface event. These conventions are consistent with other EMD documentation. The convention for naming the EMD processes is Ec

<subsystem abbreviation> meaningful name. The *Ec* identifies the process as an EMD developed process versus a Commercial Off The Shelf (COTS) product. The *subsystem abbreviations* are listed subsystem-by-subsystem.

- Cl for CLS
- Cs for CSS
- Dl for DPL
- Dm for DMS
- DPLINGEST for Data Pool Ingest (new subsystem for 7.20)
- Ds for DSS
- Ms for MSS
- Nb for SSS
- Om for OMS

The *meaningful name* identifies the process and its functionality within the subsystem, CSCI, or CSC. An example is EcDsScienceDataServer, which identifies an EMD-developed DSS process called the Science Data Server. Some names within an architecture diagram do not follow this convention because the names are COTS product names. All COTS product names are kept for simplicity and to adhere to licensing and trademark agreements. The remaining names that do not follow the naming convention are imbedded throughout the system and would require time to replace and cause operational disruptions. These names will be cleaned up during the final maintenance stages of the contract if directed by the customer.

### **Object-oriented modeling and design**

*Object-oriented modeling and design* is a new way of thinking about problems using models organized around real-world concepts. The fundamental construct is the object, which combines both data structure and behavior in a single entity. Object-oriented models are useful for understanding problems, communicating with application experts, modeling enterprises, preparing documentation and designing programs and databases.<sup>1</sup>

Superficially the term "object-oriented" means that we organize software as a collection of discrete objects that incorporate both data structure and behavior. This is in contrast to conventional programming in which data structure and behavior are only loosely connected. There is some dispute about exactly what characteristics are required by an object-oriented approach, but generally include four aspects: identity, classification, polymorphism and inheritance.<sup>11</sup> *Identity* means that data is quantized into discrete, distinguishable entities called *objects*. A paragraph in my document, a window on my workstation and a white queen in a chess game are examples of objects. Objects can be concrete, such as a file, or conceptual, such as a *scheduling policy* in a multi-processing operating system. Each object has its own inherent

---

<sup>1</sup> Object-oriented Modeling and design, James Rumbaugh et all, copyright 1991 by Prentice-Hall, Inc. ISBN 0-13-629841-9

identity. In other words, two objects are distinct even if all their attribute values (such as name and size) are identical.<sup>1</sup>

In the real world an object simply exists, but within a programming language each object has a unique *handle* by, which it can be uniquely referenced. The handle may be implemented in various ways, such as an address, array index or unique value of an attribute. Object references are uniform and independent of the contents of the objects, permitting mixed collections of objects to be created, such as a file system directory that contains both files and sub-directories.<sup>1</sup>

*Classification* means that objects with the same data structure (attributes) and behavior (operations) are grouped into a *class*. Paragraph, Window, and ChessPiece are examples of classes. A *class* is an abstraction that describes properties important to an application and ignores the rest. Any choice of classes is arbitrary and depends on the application.<sup>1</sup>

Each class describes a possibly infinite set of individual objects. Each object is said to be an instance of its class. Each instance of the class has its own value for each attribute but shares the attribute names and operations with other instances of the class. An object contains an implicit reference to its own class: it "knows what kind of a thing it is."<sup>1</sup>

*Polymorphism* means that the same operation may behave differently on different classes. The *move* operation, for example, may behave differently on the *Window* and *ChessPiece* classes. An *operation* is an action or transformation that an object performs or is subject to. *Right justify*, *display* and *move* are examples of operations. A specific implementation of an operation by a certain class is called a *method*. Because an object-oriented operator is polymorphic, it may have more than one method implementing it.<sup>11</sup>

In the real world, an operation is simply an abstraction of analogous behavior across different kinds of objects. Each object "knows how" to perform its own operations. In an object-oriented programming language, however, the language automatically selects the correct method to implement an operation based on the name of the operation and the class of the object being operated on. The user of an operation need not be aware of how many methods exist to implement a given polymorphic operation. New classes can be added without changing existing code, provided methods are provided for each applicable operation on the new classes.<sup>1</sup>

*Inheritance* is the sharing of attributes and operations among classes based on a hierarchical relationship. A class can be defined broadly and then refined into successively finer *subclasses*. Each sub-class incorporates, or *inherits* all the properties of its *super-class* and adds its own unique properties. The properties of the superclass need not be repeated. For example, *ScrollingWindow* and *FixedWindow* are subclasses of *Window*. Both subclasses inherit the properties of *Window*, such as a visible region on the screen.<sup>1</sup>

The EMD is a large, complex data storage and retrieval system used to store and retrieve large amounts of science and science-related data. The system was designed using an object oriented design approach. With so many objects and the sizes of some of them, it is necessary to have

---

<sup>1</sup> Object-oriented Modeling and design, James Rumbaugh et all, copyright 1991 by Prentice-Hall, Inc. ISBN 0-13-629841-9

some insight into the amount of memory being utilized within the EMD. The information about to be presented is a brief look at the memory management of the "key" (top ten utilized) objects within the EMD subsystems.

In this object oriented system design, objects are created and used via classes throughout the system to help perform the functions and meet the needs of the system. The objects for the EMD are very numerous, sometimes very large and cannot be provided in their entirety at this time. However, presented in the table below are the "key" objects for this system and how they are created, passed and deleted within the EMD.

### **Introduction to memory management approaches and memory usage table**

Good memory management in some applications is both important and requires significant planning and development time. Many important EMD applications are large, long running, multi-threaded, heavy memory users and therefore are prime candidates for improved memory management.<sup>1</sup>

Improper memory management can result in memory leaks, fast memory usage growth or large application footprints and random crashes. EMD servers are periodically purified for memory leaks and there is a history of progress in this area. Similar work should be expected to continue as development and maintenance continues.

Long running server like applications that are free from memory leaks can nonetheless have significant memory and Central Processing Unit (CPU) usage performance degradation. A common culprit is heap fragmentation. The repeated allocation and deallocation of memory (such as with the new and delete operators of C++) can result in a large number of unusable free blocks of memory. They are free blocks but are interspersed with non-free blocks. They become unusable since they are not contiguous (fragmented) and as time goes by, it becomes harder and harder for the OS to service requests for more memory. Such situations even lead to crashes of other, non-offending applications running in the same box.

There are strategies, tools and software to avoid both memory leaks and fragmentation. This includes but is not limited to:

- Periodic application of purification software (already an EMD practice)
- Software design, which uses dynamic memory as little as possible, such as automatic storage or COTS data structures
- Class-level memory management to allocate large chunks of memory instead of one class instance at a time ("Effective C++" by Scott Meyers and "Advanced C++" by James Coplien address this technique)
- Non-class level memory pools and
- COTS heap manager

---

<sup>1</sup> Object-oriented Modeling and design, James Rumbaugh et all, copyright 1991 by Prentice-Hall, Inc. ISBN 0-13-629841-9

Table 4-1 below is provided in case further memory management improvements are desired. Given operator or field input of seemingly inefficient memory or CPU usage, this table can be used to help target specific EMD subsystems, servers and frameworks or classes for improvement. It can be decided to apply some of the approaches at one level (e.g., on one guinea pig server or class) or perhaps experiment with changing the entire EMD C++ system with the use of a COTS heap manager. In any case, a great deal of planning and manpower is required.

**Table 4-1. Memory Management Table (1 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
DSS - SDSRV	EcDsScienceDataServer	DsSrRequest	Executes requests based on request type. Base class for DsSr<funct>Request where <funct> = Add ESDT, Insert, Search, and Generic	EcDsScienceDatas erver (class: DsSrConnectionMa ker)	Not passed. Request is immediately executed in DsSrConnecti onMaker	DsSrConnection Maker	One per DsSrRequest	Object is deleted when the DsSrRequest is completed.
	EcDsScienceDataServer	DsNrWarmStartManager	Singleton class controlling processing of asynchronous requests.	EcDsScienceDatas erver (class: DsNrManagedServ er)	not passed	EcDsScienceDat aServer	Either 1 instance or no instances for all of SDSRV.	Static singleton class. Object is deleted when the Science Data Server goes down.
	DsShSRequestReal	Clients, EcDsScienceDataServer	This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type checking.	Clients, EcDsScienceDatas erver	EcDsScience DataServer or other servers related	When request is finished or server goes down	1 per client request.	This class communicates between clients and EcDsScienceDataServer.

**Table 4-1. Memory Management Table (2 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
Clients and EcDsScienceDataServer	GIParameterList	This library is used by many subsystems to provide a general-purpose list object for storing various scalar and complex data types.	Clients and EcDsScienceDataServer	EcDsScience DataServer	Clients	EcDsScience DataServer	Could have many in each request	To group one or more GIParameter derived classes that store the various parameter types required building commands. Any GI type, including embedded GIParameterLists can be inserted into a GIParameterList, making it recursive in design.
EcDsScienceDataServer	DsGeESDT	Inherit from the public class - provides basic ESDT functionality.	EcDsScienceDataServer	EcDsScienceDataServer	not passed	EcDsScienceDataServer and other related applications	1 per granule	This class provides functionality common to all SDSRV data types.
DsSrManagedServer::DsSrStart()	DsBtSbsrvNotifier	This class is used to facilitate communication between the SDSRV and SBSRV through an event queue.	EcDsScienceDataServer	EcDsScienceDataServer [class::method = DsSrManagedServer::DsSrStart()]	not passed	EcDsScienceDataServer [class::method = DsSrManagedServer::DsSrStart()]	Configurable	Only one instance per session is created. The object goes away when the server goes down.

**Table 4-1. Memory Management Table (3 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
	DsMdMetadata	This class is used as a container class for metadata.	EcDsScienceDataServer [function = DsGeESDT::Insert()]	EcDsScienceDataServer [function = DsGeESDT::Insert()]	not passed	See comments/remarks column.	1 per DsGeESDT	When this object is instantiated, it uses the local memory manager. The object can be saved to the database if the user is executing an insert.
	DsMdCatalog	This class is used to manage catalog pools.	EcDsScienceDataServer [class::method= DsSrGenCatalogPool::DsSrGenCatalogPool() ]	DsSrGenCatalogPool::DsSrGenCatalogPool()	not passed	EcdsScienceDataServer	Depends on the configured pool size.	There are three default pools for catalogs: SEARCH, INSERT and DEFAULT. The object goes away when the server goes down.
	DsDbInterface	Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.	EcdsScienceDataServer	EcDsScienceDataServer	not passed	When EcDsScienceDataServer is down	2 per DsMsCatalog	User can connect to DB, execute SQL statements, verify connection states and disconnect from the database.

**Table 4-1. Memory Management Table (4 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
Clients	DsCIESDTReferenceCollector	Provides the primary interaction mechanism for client software.	Clients	EcDsScience DataServer	EcDsSdsrvTest or EcDsTscClientDriver or when clients go down	1 per client connection	As its name implies, it is a collector of the DsCIESDTReference object referred to as the clients "working collection", which is populated with the results of service requests such as "Insert", "Search."	
DMS	EcDmV0ToEcsGateway	GIParameterList:DeepA ssign()	Generic copy method for ECS composite class.	Used as part of query transport to SDSRV – therefore permeates through all request classes.	SDSRV (CSCI), Registry (CSC)	Where appropriate	Many	DMS code compensates for native leak by client side action
	EcDmV0ToEcsGateway	DmGwInventoryRequest	Handler for V0 Inventory Searches.	DmGwV0EcsRequestReceiver	not passed	DmGwV0EcsRequestReceiver	1 per inventory request	No significant leaks
	EcDmV0ToEcsGateway	DmGwBrowseRequest	Handler for V0 Browse Requests.	DmGwV0EcsRequestReceiver	not passed	DmGwV0EcsRequestReceiver	1 per browse request	No significant leaks, but for integrated browse may hold large amount of memory while active
	EcDmV0ToEcsGateway	DmGwProductRequest	Handler for V0 Product Requests.	DmGwV0EcsRequestReceiver	not passed	DmGwV0EcsRequestReceiver	1 per product request	No significant leaks
	EcDmV0ToEcsGateway	DmGwSpecializedCriteria	Representation of V0 SPECIALIZED_CRITERIA element.	EcDmV0ToEcsGateway	not passed	Where appropriate	Many per granule	Recursive class, potential for large memory usage.
	EcDmV0ToEcsGateway	DmGwDirectoryRequest	Handler for V0 Directory Searches.	DmGwV0EcsRequestReceiver	not passed	DmGwV0EcsRequestReceiver	1 per directory request	No significant leaks

**Table 4-1. Memory Management Table (5 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
	DmGwGranuleLevelSearch	SDSRV search client-side search object.	DmGwInventoryRequest	not passed	DmGwInventoryRequest	1 per dataset searched	No significant leaks	
	DmGwRequestReceiver	Listener / Dispatch class for EcDmV0ToEcsGateway.	DmGwManagedServer	not passed	DmGwManagerd Server	1 per process	No significant leaks	
	EcDmV0ToEcsGateway	Client class for distribution options provision.	DmGwDataset	not passed	DmGwDataset	1 per search	No significant leaks	
	EcRgDistOptions	Client class for subset options provision.	DmGwDataset	not passed	DmGwDataset	1 per search	No significant leaks	
	EcRgSubsetOptions	A class that collects the general parameters of EMD.	EcCsMtmGateway EcCsRegistry	not passed not passed	EccsMtMGateway EcCsRegistry	31 26		
CSS	EcCsMMGateway EcCsRegistry	GIParameterList	RWDBMemTable	EcCsRegistry LoadingTool	not passed not passed	EccsRegistry LoadingTool	8 2	
	EcCsRegistry LoadingTool		A Rogue Wave DB class that is a table of data residing in the program memory. After construction, an RWDBMemTable is no longer associated with a table in the database. An application can modify the data in an RWDBMemTable, but the changes are not propagated back to the database.					

**Table 4-1. Memory Management Table (6 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
	RWDBResult	A Rogue Wave DB class that represents a sequence of results whenever a database operation can potentially produce multiple SQL table expressions.	EcSelLoginProg LoadingTool EcCsRegistry	not passed not passed not passed	EcSelLoginProg LoadingTool EcCsRegistry	2 15 3		
	RWDBReader	Triggers that can cause results to be generated as a result of an INSERT, DELETE, or UPDATE statement.	A Rogue Wave DB class that provides row-by-row access to tabular data.	EcSelLoginProg EcCsRegistry LoadingTool EcCsIdNameServer	not passed not passed not passed	EcSelLoginProg EcCsRegistry LoadingTool EcCsIdNameServer	13 8 5 2	

**Table 4-1. Memory Management Table (7 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
	EccSMMGateway EccSEmailParser	DsCIESDTReferenceCollector	This class provides the primary interaction mechanism for client software. This class contains the specialized functions pertaining to management of state (the working collection on the server side) by mimicking that state on the client machine.	EccSMMGateway	not passed	EccSMMGateway	1	
MSS	EcAcOrderSrvr EcMsAcRegUserSrvr	RWDBMemTable	A Rogue Wave DB class that is a table of data residing in the program memory. After construction, an RWDBMemTable is no longer associated with a table in the database. An application can modify the data in an RWDBMemTable, but the changes are not propagated back to the database.	EcAcOrderSrvr	not passed	EcAcOrderSrvr	2	

**Table 4-1. Memory Management Table (8 of 13)**

Subsystem Name	Executable Name (M)	Key Classes (M)	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
EcAcOrderSrvr EcMsAcRegUserSrvr	RwDBResult	A Rogue Wave DB class that represents a sequence of results whenever a database operation may potentially produce multiple SQL table expressions.	EcAcOrderSrvr EcMsAcRegUserSrvr	not passed	EcAcOrderSrvr EcMsAcRegUserSrvr	13		
EcAcOrderSrvr EcMsAcRegUserSrvr MsCsSurveyMgrServ	RwDBReader	A Rogue Wave DB class that provides row-by-row access to tabular data.	EcAcOrderSrvr EcMsAcRegUserSrvr	not passed	EcAcOrderSrvr EcMsAcRegUserSrvr	16		

**Table 4-1. Memory Management Table (9 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/process name) (M)	Passed To (Executable/process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
DPLINGEST	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	DpCoAlert	Used to describe problematic conditions in DPL INGEST that require attention by the operator.	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	not passed	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	One per condition	Alerts are used to convey descriptions of problems to database and are maintained within the service for the lifetime of the problem. Deleted when alert is cleared by operator.
		DpCoMessage	Used to communicate a resource addition, modification, subtraction, suspension or resumption from the operator to the service.	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	not passed	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	One per operator action	Messages are used to communicate changes in resource state and are maintained within the service until the change has been made to the services representation of that resource.

**Table 4-1. Memory Management Table (10 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/process name) (M)	Passed To (Executable/process name)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
	DpCoResource		Used to describe a resource used by service. For example, ftp hosts, providers.	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	not passed	EcDlInPollingService EcDlInProcessingService EcDlInNotificationService	One per resource.	Created on startup or when a resource is added to the system. Deleted when a resource is removed from the system or the service is shutdown.
EcDlInPollingService								
EcDlInProcessingService								
EcDlInNotificationService								
	DplnPoller		Used to obtain PDR files from a specific polling location	EcDlInPollingService	not passed	EcDlInPollingService	One per polling location	Created on startup. Deleted when a polling location is removed or the service is shutdown.
EcDlInPollingService								
	DplnResourceCheckTimer		Used to periodically poll for changes in ingest resource properties.	EcDlInPollingService	not passed	EcDlInPollingService	One	Created on startup. Deleted on shutdown.
EcDlInPollingService								
	DplnPollingDataBase		Used to handle all interaction with ingest database	EcDlInPollingService	not passed	EcDlInPollingService	One	Created on startup. Deleted on shutdown.
EcDlInPollingService								
	DplnPDRParser		Used to parse a PDR file	EcDlInProcessingService	not passed	EcDlInProcessingService	One per PDR	Created when request is activated. Deleted when parsing has completed.
EcDlInProcessingService								

**Table 4-1. Memory Management Table (11 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
	DplnDDR	Used to represent the properties of a request during processing.	EcDlInProcessingService	not passed	EcDlInProcessingService	One per request activated	Created when request is activated. Deleted when request processing is complete.	
EcDlInProcessingService	DplnGranule	Used to represent the properties of a granule during processing.	EcDlInProcessingService	not passed	EcDlInProcessingService	One per granule in request	Created when request is activated. Deleted when request processing is complete.	
EcDlInProcessingService	DplnFile	Used to represent the properties of a file during processing.	EcDlInProcessingService	not passed	EcDlInProcessingService	One per file in a granule	Created when granule is activated. Deleted when granule processing is complete.	
EcDlInProcessingService	DplnProcessingDBInterface	Used to handle all interaction with ingest database	EcDlInProcessingService	not passed	EcDlInProcessingService	One	Created on startup. Deleted on shutdown.	
EcDlInNotificationService	EcDlAutoDispatcher	Email notification queue	EcDlInNotificationsService	not passed	EcDlInNotificationsService	One	Created on startup. Deleted on shutdown.	
EcDlInNotificationService	EcDlAutoDispatcher	File transfer notification queue	EcDlInNotificationsService	not passed	EcDlInNotificationsService	One	Created on startup. Deleted on shutdown.	
EcDlInNotificationService	DplnServerMessagesTimer	Used to periodically poll for changes in ingest resource properties.	EcDlInNotificationsService	not passed	EcDlInNotificationsService	One	Created on startup. Deleted on shutdown.	
EcDlInNotificationService	DplnNotifyPopulateTimer	Used to periodically add new notification actions from ingest database	EcDlInNotificationsService	not passed	EcDlInNotificationsService	One	Created on startup. Deleted on shutdown.	

**Table 4-1. Memory Management Table (12 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
EcdlInNotificationService	DplnNotifyRemoveCompletedActionsTimer		Used to remove completed notification actions from the ingest database	EcdlInNotificationsService	not passed	EcdlInNotificationService	One	Created on startup. Deleted on shutdown.
EcdlInNotificationService	DplnNotifyEmailAction		Used to perform the notification of ingest via email	EcdlInNotificationsService	not passed	EcdlInNotificationService	One per email notification	Created when new email notification is retrieved from database. Deleted when action has been completed.
EcdlInNotificationService	DplnNotifyFileTransferAction		Used to perform the notification of ingest via file transfer	EcdlInNotificationsService	not passed	EcdlInNotificationService	One per file notification	Created when new file transfer notification is retrieved from database. Deleted when action has been completed.
EcdlInNotificationService	DplnNotifyDatabase		Used to handle all interaction with ingest database	EcdlInPollingService	not passed	EcdlInPollingService	One	Created on startup. Deleted on shutdown.
EcdlInOdtToXml	OdtToXmtTranslator		Used to store the ringpoint section of the PDR file in xml format	EcdlInOdtToXml	not passed	EcdlInOdtToXml	One	Created on startup. Deleted on shutdown.
EcdlInOdtToXml	OdtToXmtTranslator		Used to store the PDR file	EcdlInOdtToXml	not passed	EcdlInOdtToXml	One	Created on startup. Deleted on shutdown.

**Table 4-1. Memory Management Table (13 of 13)**

Subsystem Name	Executable Name (M)	Key Classes	Description (M)	Where Created? (Executable/ process name) (M)	Passed To (Executable/ process name) (M)	Where Deleted? (process name) (M)	Number of Instances (Example – 1 per granule)	Comments/Remarks (Items of special interest. Example - Size per instantiation, never “deleted”, etc.)
<b>OMS</b>	EcoOmOrderManager	OmSrClientDb OmSrDbInterface	Handles connection and queries to the database server.	EcoOmOrderManager	Not passed	EcoOmOrderManager	One instance	The memory is deallocated when the server comes down.
	EcoOmOrderManager	OmSrDispatchQueue	Keeps track of requests for processing.	EcoOmOrderManager	Not passed	EcoOmOrderManager	Four instances	The memory is deallocated when the server comes down.
	EcoOmOrderManager	OmServer	Main encapsulating class.	EcoOmOrderManager	Not passed	EcoOmOrderManager	One instance	The memory is deallocated when the server comes down.
	EcoOmOrderManager	OmSrDistributionRequest	Stores information related to distribution requests.	EcoOmOrderManager	Not passed	EcoOmOrderManager	One instance per request	The memory is deallocated when the server comes down, or when present request is terminated in any way.
<b>CLS</b>	Not Applicable							
<b>SSS</b>	Not Applicable							
<b>Toolkit</b>	Not Applicable							

## **4.1 Data Server Subsystem Overview**

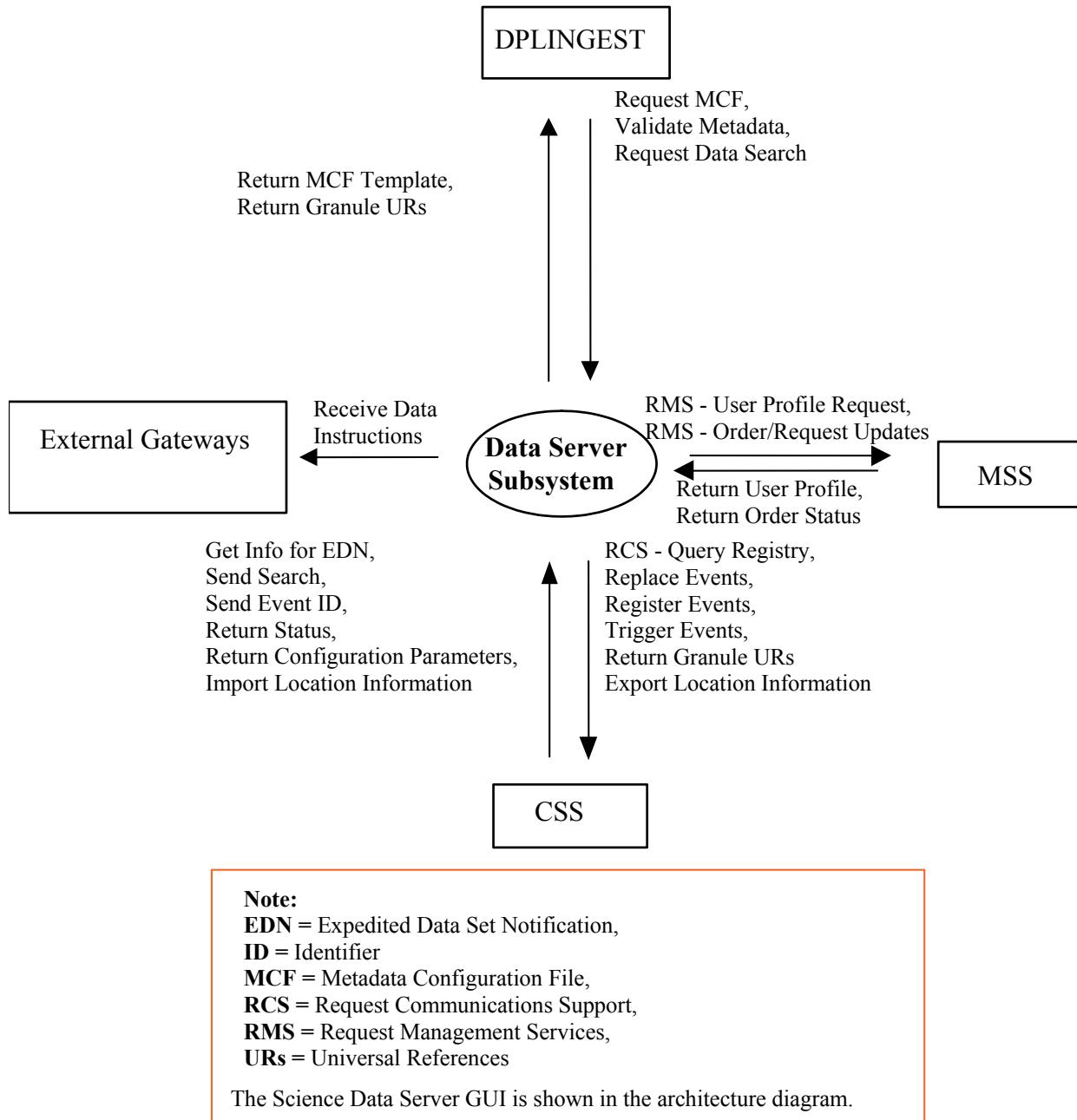
The capabilities of the Data Server Subsystem (DSS) are being reduced in release 7.20 with the removal of the SDSRV interface with the Data Distribution Server and the Storage Management. The SDSRV component of DSS continues to search, store, and manage granule and collection metadata in its repositories in support of earth science data users. Retrieval of data from DSS is now the responsibility of the Order Management Subsystem (OMS).

DSS functionality includes:

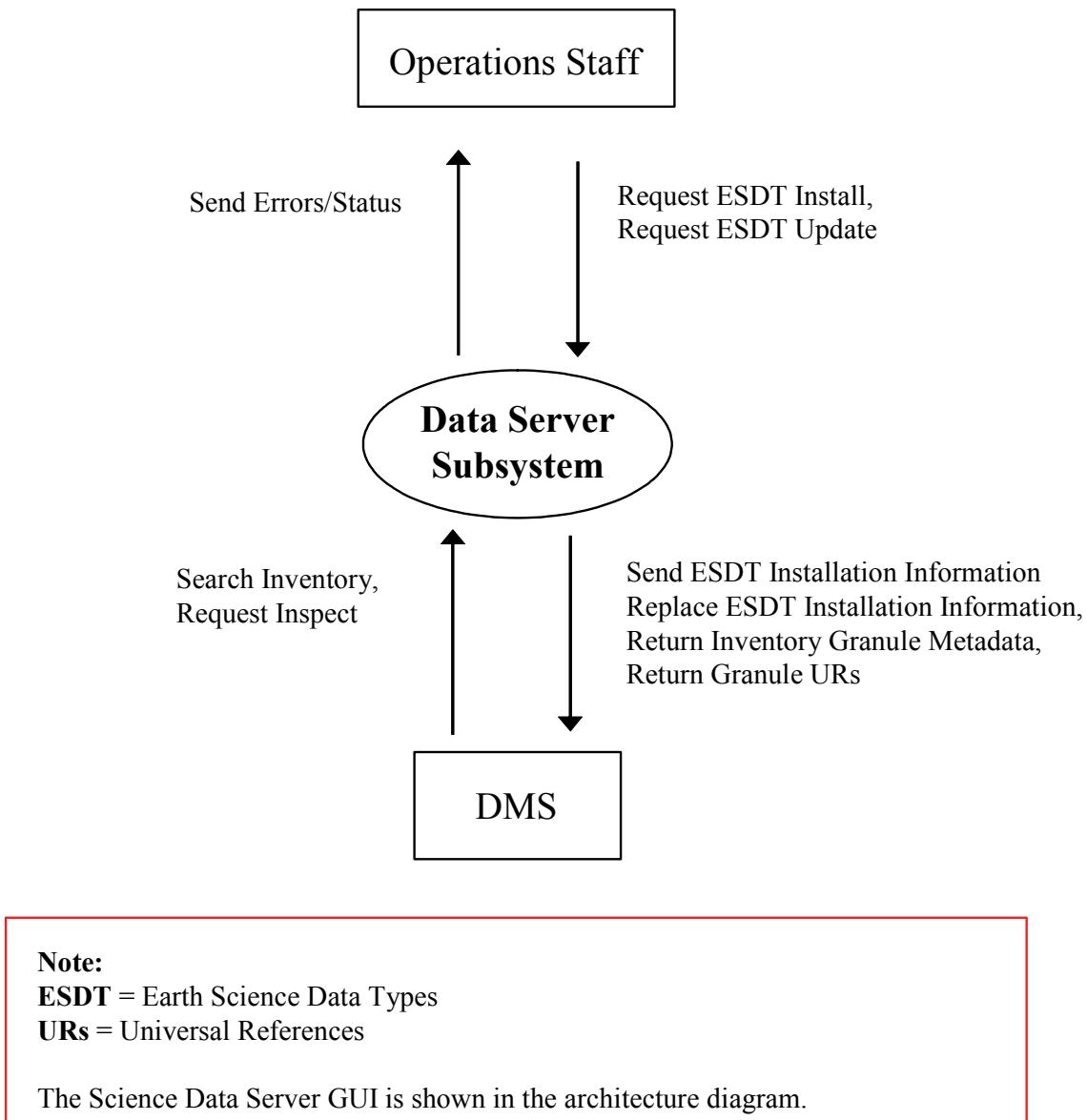
- DSS manages, validates and stores in the SDSRV database the granule and collection metadata requested by the ingest part of the Data Pool Subsystem.
- The DSS receives service requests for data and data type services from external service requesters including the Data Management Subsystem (DMS).
- The subsystem can provide references to data as a Universal Reference (UR).

### **Data Server Subsystem Context**

Figure 4.1-1 is the context diagrams for the DSS. The diagrams show the events DSS sends to other CSMS subsystems and the events sent to DSS.



**Figure 4.1-1. Data Server Subsystem Context Diagram**



**Figure 4.1-1. Data Server Subsystem Context Diagram (cont.)**

Table 4.1-1 provides descriptions of the interface events shown in the Data Server Subsystem context diagrams.

**Table 4.1-1. Data Server Subsystem Interface Events (1 of 3)**

Event	Interface Event Description
Request MCF	The <b>DPLINGEST</b> requests the Metadata Configuration File (MCF) from the DSS prior to a data insert request.
Validate Metadata	The <b>DPLINGEST</b> populates the metadata files and sends a request to the DSS to validate the metadata files.
Request Data Search	The <b>DPLINGEST</b> sends a search request to the DSS for a granule corresponding to a particular ESDT short name and version, which has a particular local granule id.
Request Management Services	The <b>MSS</b> provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services includes: <ul style="list-style-type: none"> <li>• <b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).</li> <li>• <b>User Profile Request</b> - The MSS receives requests from the DSS for user profile information such as e-mail address and shipping address from authorized users to support their processing activities.</li> <li>• </li> </ul>
Return User Profile	The <b>MSS</b> sends the user profile to the DSS for inventory searches, to request product orders, and to request subscriptions.
Return Order Status	The <b>MSS</b> provides order ids and order status information to the CLS and CSS for products requested by users.
Request Communications Support	The <b>CSS</b> provides a library of services available to each subsystem. The subsystem services required to perform specific assignments are requested from the CSS. These services include: <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• Name/Address Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> <li>• Request Distribution Media Options from the Configuration Registry</li> </ul>
Replace Events	The <b>CSS</b> Subscription Server receives by database trigger the updated subscription events with updated qualifiers for an Earth Science Data Type (ESDT) from the DSS when an ESDT is updated. This event replaces the original event in the CSS Subscription Server.
Register Events	The <b>CSS</b> Subscription Server receives the subscription events by database trigger for an Earth Science Data Type (ESDT) from the DSS when an ESDT is installed into the system or when an ESDT is updated by adding additional events.

**Table 4.1-1. Data Server Subsystem Interface Events (2 of 3)**

Event	Interface Event Description
Trigger Events	The <b>CSS</b> is notified by the DSS (via an event trigger) when a subscription event occurs on an ESDT Service.
Return Granule URs	The <b>CSS</b> and <b>DPLINGEST</b> receive Earth Science Data Type (ESDT) Universal References (URs) for the requested granules from the DSS.
Export Location Information	The DSS stores physical and logical server location information in the <b>CSS</b> .
Get Info for EDN	The <b>CSS</b> requests the EDN information from the DSS to send messages.
Send Search	The <b>CSS</b> sends requests to the DSS, on behalf of the SIPS, to get qualified granule URs returned.
Send Event ID	The <b>CSS</b> sends Event Ids to the DSS when ESDTs are installed or when ESDTs are updated by adding additional events.
Return Status	The <b>CSS</b> returns status to the DSS to simply indicate that the request was received, not that the action succeeded.
Return Configuration Parameters	The DSS receives the configuration parameters and associated values from the Registry Server within the <b>CSS</b> .
Import Location Information	The DSS retrieves physical and logical server location information from the <b>CSS</b> .
Send Distribution Notification	The <b>DPLINGEST</b> receives a distribution notification, via e-mail, from the OMS when data being distributed is to be ingested.
Return MCF Template	The <b>DPLINGEST</b> receives the MCF template to populate, from DSS, as part of the GetMCF service call.
Return Granule Urs	The <b>DPLINGEST</b> receives the Earth Science Data Type (ESDT) Universal References (Urs) for the requested granules from the DSS.
Request ESDT Install	The <b>Operations Staff</b> sends ESDT installation information to the DSS for adding descriptor, Dynamic Link Library (DLL), and version id for a new Earth Science Data Type.
Request ESDT Update	The <b>Operations Staff</b> sends updated ESDT information to the DSS for adding updated descriptor and Dynamic Link Library (DLL) information for an existing ESDT.
Request On-Demand Granule Deletion	The <b>Operations Staff</b> sends requests to the DSS to delete science granules from the archive and inventory or just the archive. The associated PH, QA and Browse granules can also be deleted.
Send ESDT Installation Information	The <b>DMS</b> Data Dictionary receives ESDT installation information from the DSS, whenever a new ESDT is installed. This data consists of Inventory and Collection level metadata.
Replace ESDT Installation Information	The <b>DMS</b> Data Dictionary receives updated ESDT information from the DSS, whenever an ESDT is updated. This data consists of updated Inventory and Collection level metadata.

**Table 4.1-1. Data Server Subsystem Interface Events (3 of 3)**

Event	Interface Event Description
Return Inventory Granule Metadata	The <b>DMS</b> receives the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the DSS.
Return Granule Urs	The <b>DMS</b> receives Earth Science Data Type (ESDT) Universal References (Urs) for the granules from the DSS.
Request Inspect	The <b>DMS</b> sends a request for an inspection of granule metadata to the DSS in support of a price estimate request.

### Data Server Subsystem Structure

The DSS is three CSCIs:

- The Science Data Server (SDSRV) CSCI manages and provides user access to collections of non-document (non hard copy) earth science data, extracts and modifies data by request, accepts browse, search, and retrieval requests from users, and catalogs data insert requests from other CSMS CSCIs, CSCs, and processes. The SDSRV CSCI manages earth science data as logical collections of related data, via interfaces independent of data formats and hardware configurations inherent in underlying storage technologies. The SDSRV manages interactive sessions with service requesters and informs the service requester of the availability of data and services via the IOS.
- The Data Server Subsystem hardware consists of the following three Hardware Configuration Items (HWCI):
  - Access Control and Management

The Access Control and Management HWCI (ACMHW) is hardware to support the Ingest and Data Server Subsystems' software to directly interact with users. The ACMHW provides a level of security by isolating other hardware items from external software access.

- Data Repository

The Data Repository HWCI (DRPHW) is hardware to provide high-capacity storage for long-term storage of data files.

- Ingest Peripherals

The Ingest Peripherals HWCI (DIPHW) is hardware to provide support to ingest and distribution via physical media.

Detailed information on hardware/software mapping, hardware diagrams, disk partitioning, etc., can be found in 920-TDx-00x, the 921-TDx-00x, and the 922-TDx-00x series of baseline documents. These documents are located at the web site <http://pete.hitc.com/baseline/index.html> and click on the Technical Documents button.

## **Use of COTS in the Data Server Subsystem**

- RogueWave's Tools.h++

The Tools.h++ class libraries provide libraries of object strings and collections. These class libraries are statically linked and delivered with the custom code installation.

- RogueWave's DBTools.h++

The DBTools.h++ C++ class libraries interact with the Sybase database Structured Query Language (SQL) server and buffer the processes from the relational database used. These class libraries are statically linked and delivered with the custom code installation.

- Rogue Wave's Net.h++

ToolsPro.h++ is a C++ class library, which includes the net.h++ class library, which provides an object-oriented interface to Inter-Process Communication (IPC) and network communication services. The Net.h++ framework enables developed code to be portable to multiple operating systems and network services.

- Integrated Computer Solutions (ICS) Builder Xcessory

The Builder Xcessory GUI builder tool modifies the displays. The Builder Xcessory generates the C++ code to produce the Maintenance Tool (Mtool) display at run time. There is no operational part of the Builder Xcessory needed at run-time.

- Sybase Adaptive Enterprise Server (ASE)

The Sybase ASE provides the capabilities to insert, update and delete database contents. The Sybase ASE must be operational to execute search and insert requests for metadata.

- Boeing Autometric's Spatial Query Server

The Spatial Query Server (SQS) provides the capability to store and search spatial metadata. SQS has spatial indexing to search on spatial metadata for the SDSRV.

- Sybase Open Client / CT\_LIB

The Sybase Open Client provides access between DSS custom code and the Sybase ASE DBMS.

- University of Illinois' Hierarchical Data Format (HDF)

HDF provides EOS extended capabilities for sub-setting services with the SDSRV CSCI.

- University of Colorado's Object Description Language (ODL)

ODL provides a general architecture, independent means of passing metadata files between subsystems.

- CCS Middleware Client

CCS Middleware Client provides DSS with communications between other subsystems. CCS Middleware can reside on one or both sides of the interface. An instance must be installed on the platform where DSS resides. Although the CCS Middleware Client is part of CSS, this COTS product must be installed for DSS to run in the operational and test environment.

## Error Handling and processing

EcUtStatus is a class used throughout the EMD custom code for general error reporting. It is almost always used as a return value for functions and allows detailed error codes to be passed back up function stacks.

DsShError is a Science Data Server specific class used mainly for exception handling.

DsShErrorDetails is a Science Data Server class that can be used to convert error details (in an EcUtStatus object) into more meaningful text messages.

The Science Data Server uses two main mechanisms for error handling.

### 1. Return Values

Functions can return an EcUtStatus object, which can be used to indicate a general success/failure status. Also, more detailed information on the exact reason for the failure can be provided. For example, a granule cannot be acquired because it has restricted access privileges. This is the most widely used mechanism within the Science Data Server and in general these errors get propagated back up to the top-level functions with ALOG error messages being generated along the way.

### 2. Exceptions

Some functions (for example, class constructors) cannot return values to indicate success or failure. These functions may throw exceptions, usually instances of the DsShError class. These errors are usually caught by other functions at a low level and converted into EcUtStatus return values (as described in 1).

In addition, the DsShErrorDetails class can be used to map error values (as contained in an EcUtStatus object) into text messages. This enables better reporting of errors in the Science Data Server logs.

Currently, the Science Data Server client interface only supports returning error messages back to client programs, along with a generic success/failed status.

For writing messages to the Applications Log (ALOG), the following functions are used:

DsLg.LogError sends a message to the ALOG at severity level 1. For example, DsLg.LogError (“DsMdMetadataCheckpoint1”, “Bad granule UR”);

DsLg.LogWarning sends a message to the ALOG at severity level 2. For example, DsLg.LogWarning (“DsMdMetadataCheckpoint2”, “Unable to retrieve granule metadata”);

DsLgLogInformational sends a message to the ALOG at severity level 3. For example, DsLgLogInformational (“DsMdMetadataCheckpoint3”, “Failed to construct granule”);

For writing messages to the debug log, the following macros are used:

PF\_STATUS writes a message at a “log level” of 1 to the debug log. For example, PF\_STATUS  
{cerr << “Issue rpc to DMS” << endl;}

PF\_VERBOSE writes a message at a “log level” of 2 to the debug log. For example, PF\_VERBOSE {cerr << “Request received from client” << endl;}

PF\_DEBUG writes a message at a “log level” of 3 to the debug log. For example, PF\_DEBUG  
{cerr << “Saved request to database” << endl;}

The class EcUtStatus is used to hold the actual error number. The EcUtStatus object is returned to the SDSRV clients when the request is complete.

The DDICT CSCI uses two main mechanisms for error handling.

## 1. Return Values

Functions can return an EcUtStatus object, which can be used to indicate a general success/failure status. Also, more detailed information on the exact reason for the failure can be provided. This is the most widely used mechanism within the DDICT and in general these errors get propagated back up to the top-level functions with ALOG error messages being generated along the way.

## 2. Exceptions

Some functions (for example, class constructors) cannot return values to indicate success or failure. These functions may throw exceptions, usually instances of the DsShError class. These errors are usually caught by other functions at a low level and converted into EcUtStatus return values (as described in 1).

In addition, the DsShErrorDetails class can be used to map error values (as contained in an EcUtStatus object) into text messages.

For writing messages to the Applications Log (ALOG), the following functions are used:

EcLg.LogError sends a message to the ALOG at severity level 1. For example,

```
catch (DsShError& err)
{ EcLg.LogError (“DsDdSchedulerExecuteFunctionError” , 0, err.GetMsg());}
```

EcLg.LogInformational sends a message to the ALOG at severity level 3. For example,  
EcLg.LogInformational (“ConfigVarMissing”,status.GetLogMessageLink(),  
“EcCUtRpcClientIDConfigTag var not set in Config File.”);

For writing messages to the debug log, the following macros are used:

PF\_STATUS writes a message at a “log level” of 1 to the debug log. For example, PF\_STATUS  
{cerr << “DsDdCBCache::Create Creating first instance.” << endl;}

PF\_VERBOSE writes a message at a “log level” of 2 to the debug log. For example,  
PF\_VERBOSE {cerr << “Calling DsStArchive::Create. ReqID | FullArchiveID:\n”  
“<< myRequestID <<” | “<< FullArchiveID << “ and\n”  
“<< “RpcID = “ << RpcId\_CR.AsString() << endl;}

PF\_DEBUG writes a message at a “log level” of 3 to the debug log.

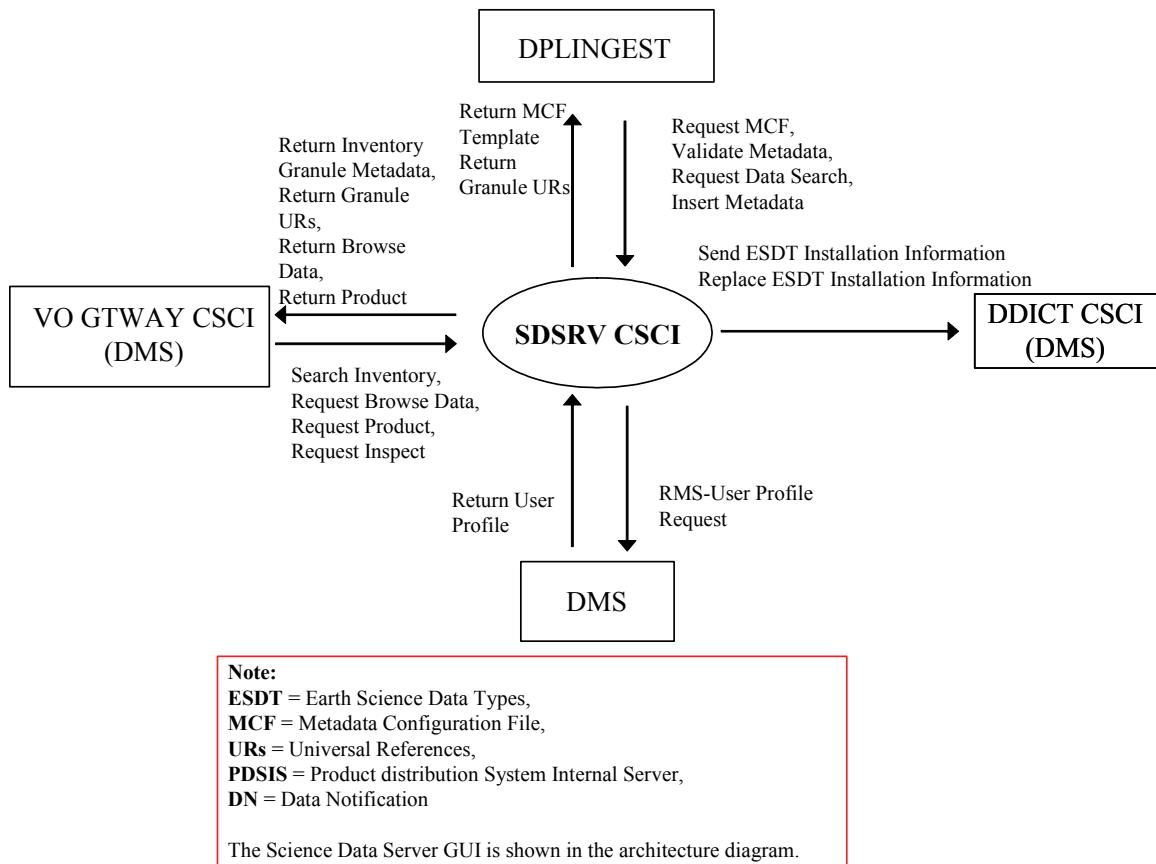
#### **4.1.1 Science Data Server Software Description**

##### **4.1.1.1 Science Data Server Functional Overview**

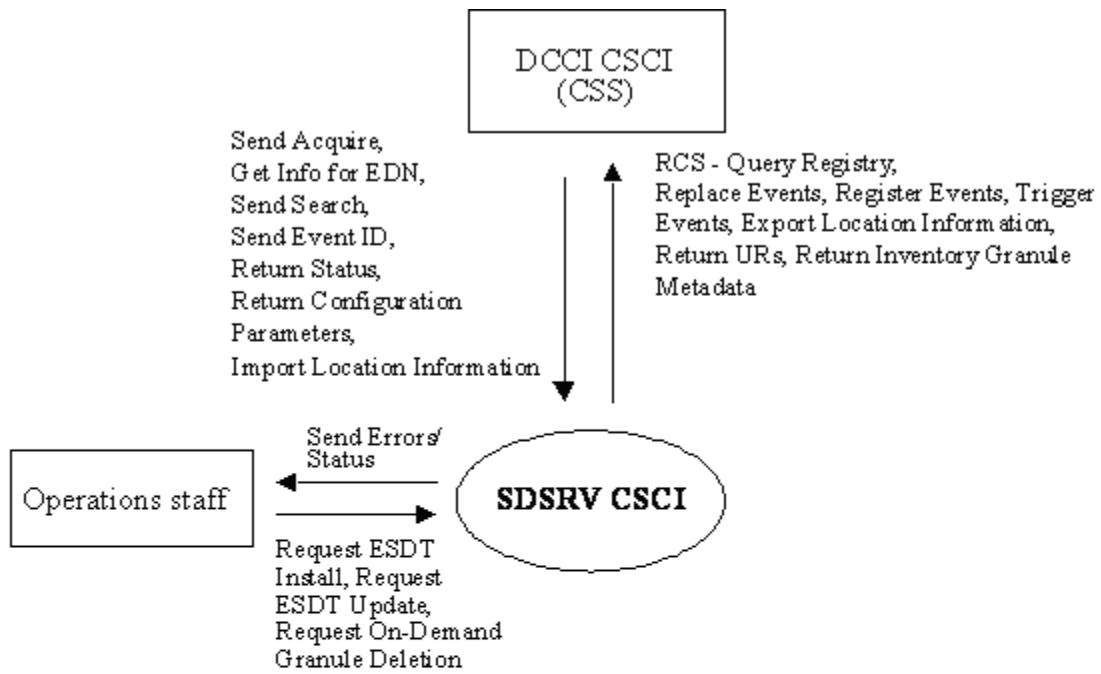
The SDSRV CSCI provides a catalog of Earth Science Data holdings, and the Earth Science Data Type services that operate on the data. The SDSRV CSCI provides a catalog of metadata describing the archived data holdings of the subsystem and provides mechanisms to acquire the data from the archive. The SDSRV CSCI also provides data type services on the catalog and a data reduction or sub-setting and reformatting services.

##### **4.1.1.2 Science Data Server Context**

Figure 4.1-2 the SDSRV CSCI context diagrams. The diagrams show the events sent to the SDSRV CSCI and the events the SDSRV CSCI sends to other CSCIs. The events have been grouped by CSCI



**Figure 4.1-2. SDSRV CSCI Context Diagram**



**Note:**

RCS = Request Communications Support,

EDN = Expedited Data Set Notification,

ESDT = Earth Science Data Type,

ID = Identifier

The Science Data Server GUI is shown in the architecture diagram.

**Figure 4.1-2. SDSRV CSCI Context Diagram (cont.)**

Table 4.1-2 provides descriptions of the interface events shown in the SDSRV CSCI context diagrams.

**Table 4.1-2. SDSRV CSCI Interface Events (1 of 2)**

Event	Interface Event Description
Request MCF	The <b>DPLINGEST CSCI</b> requests the Metadata Configuration File (MCF) template, from the SDSRV CSCI, for each input or output data type, respectively, prior to a data insert request. The SDSRV CSCI provides the MCF information as part of the GetMCF service call.
Validate Metadata	The <b>DPLINGEST CSCI</b> populates the metadata files and sends a request to the SDSRV CSCI to validate the metadata files.
Request Data Search	The <b>DPLINGEST CSCI</b> sends a search request to the SDSRV CSCI for a granule corresponding to a particular ESDT short name and version, which has a particular local granule id.
Ingest Granules	The <b>DPLINGEST CSCI</b> sends requests to the SDSRV CSCI to insert metadata for a particular file or files which will be stored in the DPL archive. The metadata is catalogued in the SDSRV inventory as a granule of a particular ESDT short name and version. For the DPLINGEST CSCI, this data can be algorithms, Level 0 (L0) data, standard products, ancillary data, correlative data or calibration data.
Send ESDT Installation Information	The <b>DDICT CSCI</b> receives ESDT installation information from the SDSRV CSCI, whenever a new ESDT is installed. This data consists of Inventory and Collection level metadata.
Replace ESDT Installation Information	The <b>DDICT CSCI</b> receives updated ESDT information from the SDSRV CSCI, whenever an ESDT is updated. This data consists of Inventory and Collection level metadata. The updated information replaces the ESDT information in the DDICT CSCI.
Search Inventory	The <b>V0 GTWAY CSCI</b> sends inventory search requests to the SDSRV CSCI to search inventory (metadata).
Request Inspect	The <b>V0 GTWAY CSCI</b> sends a request for an inspection of granule metadata to the SDSRV CSCI in support of a price estimate request.
Return MCF Template	The <b>DPLINGEST CSCI</b> receives the template to populate as part of the GetMCF service call to the SDSRV CSCI.
Request MCF	The <b>AITTL CSCIs</b> request the Metadata Configuration File (MCF), from the SDSRV CSCI, for each input or output data type, respectively, prior to a data insert request. The SDSRV CSCI provides the MCF information as part of the GetMCF service call.
Request Management Services	The <b>MCI</b> provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: <ul style="list-style-type: none"> <li>• <b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).</li> <li>• <b>Request User Profile</b> - The MCI receives requests from the SDSRV CSCI for user profile information such as e-mail address and shipping address from authorized users to support their processing activities.</li> </ul>

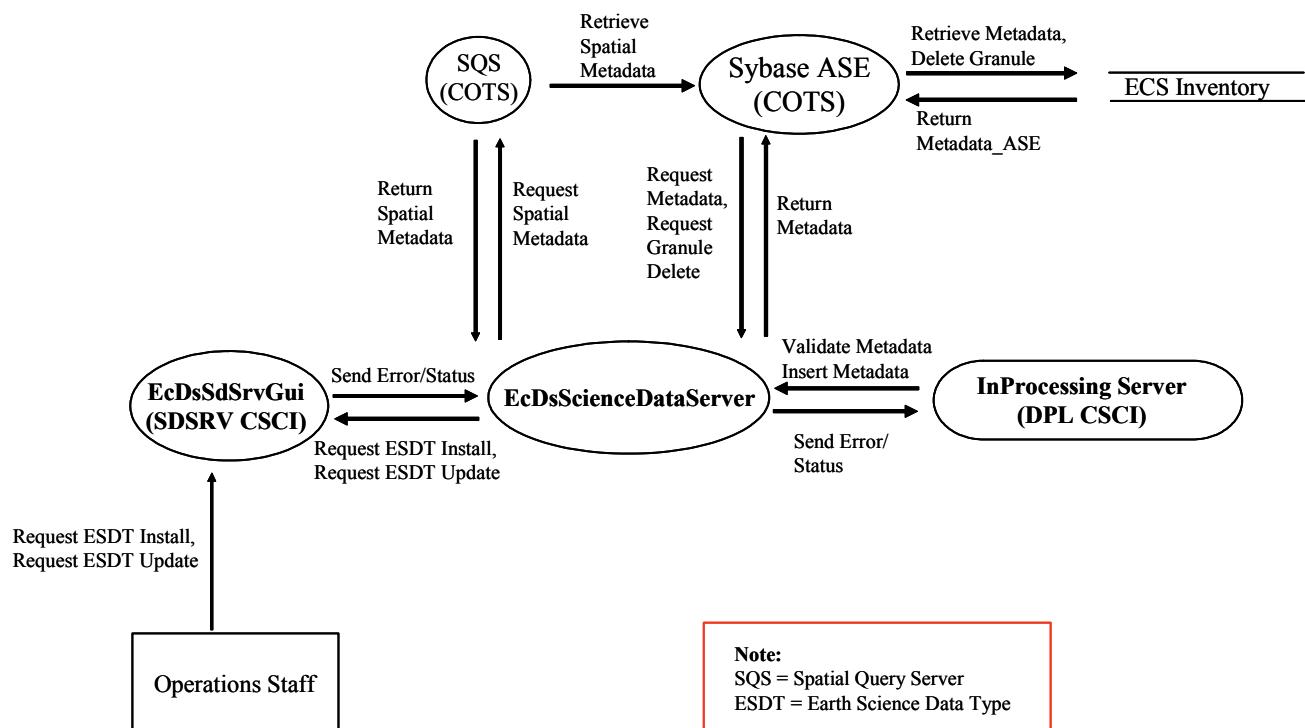
**Table 4.1-2. SDSRV CSCI Interface Events (2 of 2)**

Event	Interface Event Description
Request Communications Support (RCS)	<p>The <b>DCCI CSCI</b> provides a library of services available to each CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include:</p> <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• Name/Address Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> <li>• Request Distribution Media Options from the Configuration Registry</li> </ul>
Replace Events	<p>The <b>DCCI CSCI</b> (Subscription Server) receives updated subscription events with modified qualifiers for an Earth Science Data Type (ESDT) from the SDSRV CSCI, when an ESDT is updated. This event replaces the original event in the DCCI CSCI.</p>
Register Events	<p>The <b>DCCI CSCI</b> receives the subscription events for an Earth Science Data Type from the SDSRV CSCI, when an ESDT is installed into the system or when an ESDT is updated by adding additional events.</p>
Trigger Events	<p>The <b>DCCI CSCI</b> receives notification from the SDSRV CSCI (via an event trigger) when a subscription event occurs on an Earth Science Data Type Service.</p>
Export Location Information	<p>The SDSRV CSCI stores physical and logical server location information in the <b>DCCI CSCI</b>.</p>
Return URs	<p>The <b>DCCI CSCI</b> receives Earth Science Data Type (ESDT) Universal References (URs) for the requested granules from the SDSRV CSCI.</p>
Request ESDT Install	<p>The <b>Operations Staff</b> sends ESDT installation information to the SDSRV CSCI for adding the descriptor, Dynamic Link Library (DLL), and version id for a new Earth Science Data Type.</p>
Request ESDT Update	<p>The <b>Operations Staff</b> sends updated ESDT information to the SDSRV CSCI for adding updated descriptor and Dynamic Link Library (DLL) information for an existing ESDT.</p>
Request On-Demand Granule Deletion	<p>The <b>Operations Staff</b> sends requests to the SDSRV CSCI to delete science granules from the archive and inventory or just the archive. The associated PH, QA and Browse granules can also be deleted.</p>
Get Info for EDN	<p>The <b>DCCI CSCI</b> receives the Expedited Data Set Notification (EDN) information from the SDSRV CSCI, by request, and uses the EDN information to send messages.</p>
Send Search	<p>The <b>DCCI CSCI</b> sends requests to the SDSRV CSCI, on behalf of the SIPS, to get qualified granule URs returned.</p>
Return Configuration Parameters	<p>The SDSRV CSCI receives the configuration parameters and associated values from the Registry Server from the <b>DCCI CSCI</b>.</p>
Import Location Information	<p>The SDSRV CSCI retrieves physical and logical server location information from the <b>DCCI CSCI</b>.</p>

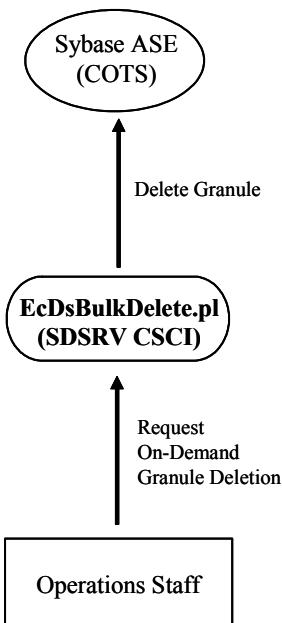
#### 4.1.1.3 Science Data Server Architecture

Figure 4.1-3 is the SDSRV CSCI architecture diagrams. The diagrams show the events sent to the SDSRV CSCI processes and the events the SDSRV CSCI processes send to other processes.

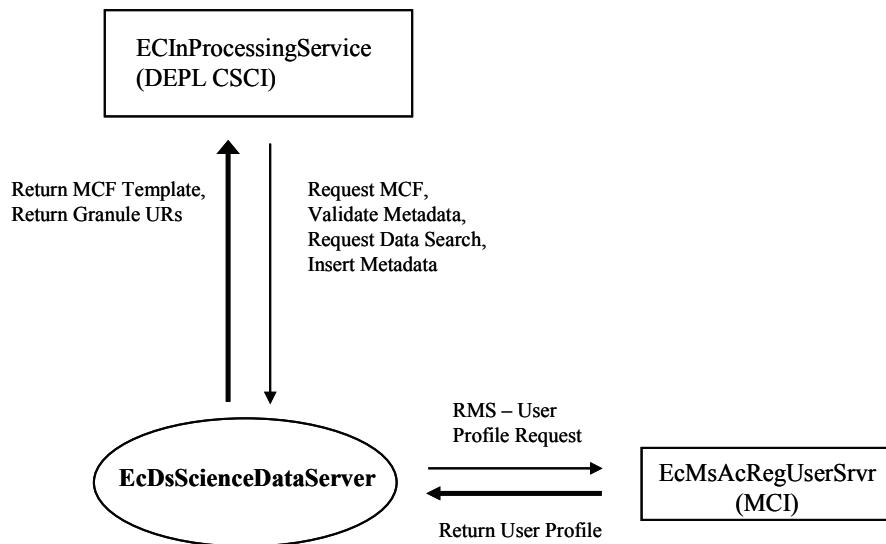
The Science Data Server (SDSRV) CSCI is five processes: three custom developed processes and two COTS processes. The three custom developed processes are the Science Data Server (EcDsScienceDataServer), the Science Data Server GUI (EcDsSdSrvGui), and the Granule Deletion Administration Tool (EcDsBulkDelete.pl). The COTS processes are the Sybase ASE and the Spatial Query Server (SQS). The SDSRV CSCI uses the Sybase ASE Database Management System (DBMS) for Inventory and Configuration data storage. The server holds Earth Science Data Type configuration information and the data catalog for all the archived products found at a DAAC.



**Figure 4.1-3. SDSRV CSCI Architecture Diagram**

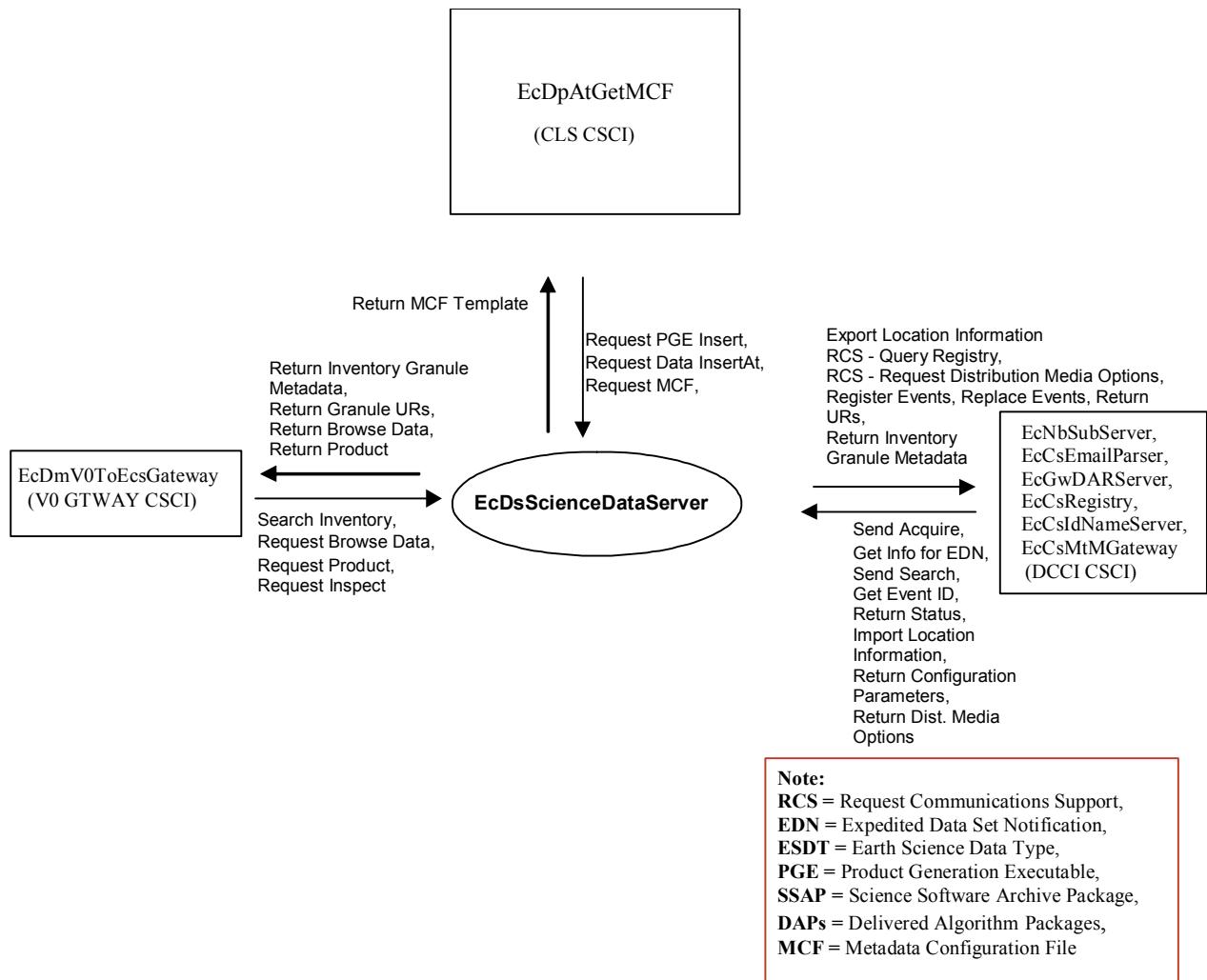


**Figure 4.1-3. SDSRV CSCI Architecture Diagram (cont.)**



**Note:**  
**RMS** = Request Management Services,  
**MCF** = Metadata Configuration File,  
**URs** = Universal References,  
**MCI** = Management Software CSCI

**Figure 4.1-3. SDSRV CSCI Architecture Diagram (cont.)**



**Figure 4.1-3. SDSRV CSCI Architecture Diagram (cont.)**

#### 4.1.1.4 Science Data Server Process Descriptions

Table 4.1-3 provides descriptions of the processes shown in the SDSRV CSCI architecture diagrams.

**Table 4.1-3. SDSRV CSCI Processes (1 of 2)**

Process	Type	Hardware CI	COTS / Developed	Functionality
EcDsScienceDataServer	Server	ACMHW	Developed	<p>The EcDsScienceDataServer server manages collections of earth science and related data, and service requests for the storage, search, retrieval, and manipulation of data within those collections. The science data server performs the following functions:</p> <ul style="list-style-type: none"> <li>• Manages earth science data as logical collections of related data, using interfaces independent of any data formats and hardware configurations provided by underlying storage technologies,</li> <li>• Manages interactive sessions with users,</li> <li>• Manages the processing of search service requests from the DMS (V0 Gateway) as part of distribution through OMS</li> <li>• Manages the processing of service requests from the DPLINGEST "insert", validate, and access metadata for long-term storage and access in the DPL,</li> <li>• Provides subscription events and event triggers to the CSS subscription server</li> <li>• Issues event information to the CSS EcSbSubServer via the SDSRV database</li> </ul> <p>The EcDsScienceDataServer supports:</p> <ul style="list-style-type: none"> <li>• Single requests, one at a time</li> <li>• Multiple concurrent requests</li> <li>• Asynchronous request processing</li> <li>• Request processing buffered from SOCKET Communication Call threads</li> <li>• Multiple threads within a single user session</li> </ul>
EcDsSdSrvGui	GUI	ACMHW	Developed	<p>The EcDsSdSrvGui provides an operator interface for:</p> <ul style="list-style-type: none"> <li>• Receiving descriptor files and dynamic link libraries (dll) for configuring ESDTs into the EcDsScienceDataServer</li> <li>• Monitoring active EcDsScienceDataServer requests</li> <li>• Updating ESDT information in the EcDsScienceDataServer</li> </ul> <p>The EcDsSdSrvGui supports:</p> <ul style="list-style-type: none"> <li>• Single requests, one at a time</li> </ul>

**Table 4.1-3. SDSRV CSCI Processes (2 of 2)**

Process	Type	Hardware CI	COTS / Developed	Functionality
EcDsBulkDelete.pl	Command Line Utility	ACMHW	Developed	The EcDsBulkDelete.pl provides a command line operator interface for deleting granules in the EcDsScienceDataServer.
Sybase ASE	Server	ACMHW	COTS	Provides the management of spatial data types of an earth science catalog of metadata. Includes capabilities for searching and storing the catalog.
Spatial Query Server (SQS)	Server	ACMHW	COTS	Provides the capability to manage spatial data types of earth science catalog metadata.

#### **4.1.1.5 Science Data Server Process Interface Descriptions**

Table 4.1-4 provides descriptions of the interface events shown in the SDSRV CSCI architecture diagram for Figure 4.1-3.

**Table 4.1-4. SDSRV CSCI Process Interface Events (1 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Metadata	One per request to store, search, delete, update, Earth Science Metadata	Process: Sybase ASE (COTS)	Processes: EcDsScienceDataServer Library: DsDb Class: DsDbInterface via the Sybase ASE, SQS, and Sybase Open Client COTS	The <b>Sybase ASE</b> receives requests from the EcDsScienceDataServer to store, search, delete, or update Earth Science Metadata. The results are sent back to the EcDsScienceDataServer. The M&O Staff must manually change a configured parameter in order for the EcDsScienceDataServer to communicate directly with the Sybase ASE.

**Table 4.1-4. SDSRV CSCI Process Interface Events (2 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Granule Delete	One granule per request	<i>Process:</i> Sybase ASE (COTS)	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDb <i>Class:</i> DsDbInterface	The <b>Sybase ASE</b> receives requests to delete granules from the EcDsScienceDataServer.
Retrieve Metadata	One to many per metadata request	Data Tables within the EMD Inventory	<i>Process:</i> Sybase ASE <i>Libraries (Sybase):</i> Libtcl.so Libtli.so Libsybdb.so <i>Class:</i> Sybase Open Client /ct_lib	The <b>Sybase ASE</b> retrieves metadata from the EMD Inventory database and returns the metadata to the SQS or the EcDsScienceDataServer.
Delete Granule	One per request	Data Tables within the EMD Inventory	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDb <i>Class:</i> DsDbInterface	The <b>Sybase ASE</b> requests granules to be deleted from the EMD Inventory data tables.
Return Metadata_ASE	One to many per metadata request	<i>Process:</i> Sybase ASE (COTS)	Data Tables within the EMD Inventory	The <b>EMD Inventory data tables</b> provide the metadata or spatial metadata to the Sybase ASE.
Return Metadata	Per request	<i>Process:</i> Sybase ASE (COTS)	<i>Process:</i> EcDsScienceDataServer <i>Libraries:</i> DsDb <i>Class:</i> DsDbInterface	Metadata results are sent back to the EcDsScienceDataServer via the SQS. The results include the status of the Sybase ASE commands.
Request ESDT Install	One per new ESDT installation	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsAd <i>Class:</i> DsAdDataTypeCollector	Operations Staff <i>Process:</i> EcDsSdSrvGui <i>Class:</i> DsGuSdDatatype	The <b>Operations Staff</b> sends ESDT installation information for adding the descriptor, dynamic link library (dll), and version ID for a new ESDT to the EcDsScienceDataServer, via the EcDsSdSrvGui.

**Table 4.1-4. SDSRV CSCI Process Interface Events (3 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request ESDT Update	One per new ESDT update	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsAd <i>Class:</i> DsAdDataTypeCollector	Operations Staff <i>Process:</i> EcDsSdSrvGui <i>Class:</i> DsGuSdDatatype	The <b>Operations Staff</b> sends updated ESDT information, via the EcDsSdSrvGui, to the EcDsScienceDataServer, for adding updated descriptor and dynamic link library (dll) information for an existing ESDT.
Return Spatial Metadata	Per request	<i>Process:</i> EcDsScienceDataServer <i>Libraries:</i> DpPrDssIF, DsDb <i>Class:</i> DsDbInterface	<i>Process:</i> SQS (COTS)	Metadata results are sent back to the EcDsScienceDataServer via the SQS. The results include the status of the Sybase ASE and SQS Server commands. The SQS handles the translation of spatial metadata data types (understood by the EcDsScienceDataServer and SQS) to relational data types (understood by SQS and the Sybase ASE) and vice versa.
Retrieve Spatial Metadata	One per request	<i>Process:</i> Spatial Query Server (SQS) [COTS]	<i>Processes:</i> EcDsScienceDataServer <i>Library:</i> DsDb <i>Class:</i> DsDbInterface via the Sybase ASE, SQS, and Sybase Open Client COTS	The <b>SQS</b> sends requests to the Sybase ASE to store, search, delete, or update Earth Science Metadata. The results are sent back to the SQS. The default configuration is for the EcDsScienceDataServer to communicate via the SQS to the EMD inventory.

**Table 4.1-4. SDSRV CSCI Process Interface Events (4 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Spatial Metadata	One per request to store, search, delete, or update spatial Earth Science Metadata.	<i>Process:</i> SQS (COTS)	<i>Process:</i> EcDsScienceDataServer <i>Libraries:</i> DpPrDssIF, DsDb <i>Class:</i> DsDbInterface	The <b>SQS</b> receives requests from the EcDsScienceDataServer to store, search, delete, or update spatial Earth Science Metadata in the EMD inventory database.
Request On-Demand Granule Deletion	Per Operations Staff request	<i>Process:</i> Sybase ASE (COTS)	Operations Staff <i>Process:</i> EcDsBulkDelete.pl	The <b>Operations Staff</b> sends requests, via the <b>Granule Delete Tool</b> , to the Sybase ASE database server to delete science granules from the archive and inventory or just the archive. The associated PH, QA and Browse granules can also be deleted. A physical delete or a Delete From Archive can be performed.
Delete Granule	One per request	<i>Script:</i> EcDsBulkDelete.pl	<i>Process:</i> Sybase ASE (COTS)	The Sybase ASE database server deletes the requested granule from the archive and inventory or just the archive.
Request MCF	One per set of external data received by EMD	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Class:</i> DsCIDescriptor	<i>Process:</i> EcInProcessingService <i>Library:</i> InPreprocess <i>Class:</i> InDataPreprocessTask	The <b>EcInProcessingService</b> process requests the Metadata Configuration File (MCF) from the EcDsScienceDataServer, prior to a data insert request. The EcDsScienceDataServer provides the MCF information as part of the GetMCF service call.

**Table 4.1-4. SDSRV CSCI Process Interface Events (5 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Validate metadata	One per data insert request.	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Class:</i> DsCIDescriptor	<i>Process:</i> EcInProcessingService <i>Library:</i> InPreprocess <i>Class:</i> InDataPreprocessTask	The <b>EcInProcessingService</b> process populates the metadata files and sends requests to the EcDsScienceDataServer to validate the metadata files.
Request Data Search	One per input pointer in metadata or per granule pointer in linkage file	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Class:</i> DsCIQuery	<i>Process:</i> EcInGran <i>Library:</i> InPreprocess <i>Class:</i> InDataPreprocessTask	The <b>EcInProcessingService</b> process sends a search request to the EcDsScienceDataServer for a granule corresponding to particular ESDT short name and version, which has a particular local granule id.
Ingest Granules	One per data insert request from EcInGran	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIRequest, DsCICommand, DsGeESDT	<i>Process:</i> EcInProcessingService <i>Library:</i> InPreprocess <i>Class:</i> InDataServerInsertionTask	The <b>EcInProcessingService</b> process sends requests to the EcDsScienceDataServer to insert catalog a granule's metadata in the SDSRV inventory as a granule of a particular ESDT short name and version. This is the metadata for algorithms, Level 0 (L0) data, standard products, ancillary data, correlative data or calibration data.

**Table 4.1-4. SDSRV CSCI Process Interface Events (6 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services (RMS)	One service per request	N/A	N/A	The <b>EcMsAcRegUserSrvr</b> provides a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI process Framework. The basic management library of services includes: (See the table cells below.)
RMS (cont.)	At system startup or shutdown and for restarts	<i>Process:</i> EcMsAcRegUserSrvr	DAAC unique startup scripts	<b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625), identified in Section 2.2.1 of this document.
RMS (cont.)	One profile per request	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Class:</i> MsAcUsrProfile	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsSr <i>Class:</i> DsSrManagedServer	<b>User Profile Request</b> - The EcMsAcRegUserSrvr receives requests from the EcDsScienceDataServer for user profile information such as e-mail address and shipping address from authorized users to support their processing activities.
Return User Profile	One per request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsSr <i>Class:</i> DsSrManagedServer	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Classes:</i> MsAcUsrProfile, RWPortal	The <b>EcMsAcRegUserSrvr</b> returns the user profile to the EcDsScienceDataServer.

**Table 4.1-4. SDSRV CSCI Process Interface Events (7 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Data InsertAt	One per data insert request	<p><i>Process:</i> EcDsScienceDataServer</p> <p><i>Library:</i> DsCI</p> <p><i>Classes:</i> DsCIRequest, DsCICommand, DsGeESDT</p>	<p><i>Processes:</i> EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile</p> <p><i>Library:</i> DpAtDsrv</p> <p><i>Class:</i> DpAtDsrv</p>	The <b>EcDpAtInsertExeTarFile</b> , <b>EcDpAtInsertStaticFile</b> , and <b>EcDpAtInsertTestFile</b> , processes send requests to the EcDsScienceDataServer to insert a particular file or files into the archive. Inserted data is accompanied by metadata. The metadata is catalogued in the SDSRV inventory as a granule of a particular ESDT short name and version. For the EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, and EcDpAtInsertTestFile processes these files can be processing output.
Request MCF	One per set of external data received by the EMD	<p><i>Process:</i> EcDsScienceDataServer</p> <p><i>Library:</i> DsCI</p> <p><i>Class:</i> DsCIDescriptor</p>	<p><i>Processes:</i> EcDpAtGetMCF</p> <p><i>Library:</i> DpPrDssIF</p> <p><i>Class:</i> DpPrDSSInterface</p>	The <b>EcDpAtGetMCF</b> processes request the Metadata Configuration File (MCF) from the EcDsScienceDataServer, prior to a data insert request.
Export Location Information	One per server	<p><i>Process:</i> EcCsIdNameServer</p> <p><i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce</p> <p><i>Classes:</i> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy</p>	<i>Process:</i> EcDsScienceDataServer	The EcDsScienceDataServer places physical and logical location information in the <b>EcCsIdNameServer</b> .

**Table 4.1-4. SDSRV CSCI Process Interface Events (8 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	One service per request	<p><i>Process:</i> EcCsIdNameServer</p> <p><i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce</p> <p><i>Classes:</i> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy</p> <p><i>Library (Common):</i> EcUr</p> <p><i>Class:</i> EcUrServerUR</p> <p><i>Library:</i> event</p> <p><i>Class:</i> EcLgErrorMsg</p> <p><i>Process:</i> EcSbSubServer</p> <p><i>Library:</i> EcSbCl</p> <p><i>Classes:</i> EcCIEvent, EcCITriggerEventCb, EcCIRegisterEventCb</p> <p><i>Process:</i> EcCsEmailParser</p> <p><i>Class:</i> EcCsEmailParser</p> <p><i>Process:</i> EcCsRegistry</p> <p><i>Library:</i> EcCsRegistry</p> <p><i>Class:</i> EcRgRegistryServer_C</p>	<p><i>Process:</i> EcDsScienceDataServer</p> <p><i>Libraries:</i> DsDe1, DsBtSh</p> <p><i>Classes:</i> DsDeEventCustomizer, DsDeServiceCustomizer, DsBtSbSbrvNotifier</p>	<p>The DCCI CSCI provides a library of services available to each CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include:</p> <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• Name/Address Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> <li>• Request Distribution Media Options from the Configuration Registry (when it first accesses an Earth Science Data Type.)</li> </ul>

**Table 4.1-4. SDSRV CSCI Process Interface Events (9 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Register Events	One per ESDT installation	<i>Process:</i> Sybase ASE (COTS)	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeEventCustomizer	The EcDsScienceDataServer increments and retrieves the next Event Id from the SDSRV DB instead of the EcNbSubServer for the subscription events for an Earth Science Data Type to the <b>EcSbSubServer</b> when an ESDT is installed into the system or when an ESDT is updated by adding additional subscription events.
Replace Events	One per ESDT update	<i>Process:</i> EcSbSubServer <i>Library:</i> EcSbSrSh <i>Class:</i> EcCIEvent	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeEventCustomizer	The EcDsScienceDataServer increments and retrieves the next Event Id from the SDSRV DB instead of the EcNbSubServer for an Earth Science Data Type (ESDT) to the <b>EcSbSubServer</b> when an ESDT is updated in the system. This replaces the previous information.
Trigger Events	One per subscription event	<i>Process:</i> EcSbSubServer <i>Library:</i> EcSbCI <i>Class:</i> EcCIEvent	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsBtSh <i>Class:</i> DsBtSbsrvNotifier	The EcDsScienceDataServer notifies the <b>EcNbSubServer</b> (via an event trigger) when a subscription event occurs on an Earth Science Data Type Service.
Return URs	One per query	<i>Process:</i> EcCsMtMGateway <i>Classes:</i> EcCsMtMDataServerMgr, EcCsMtMECSOrderProxy ,	<i>Process:</i> EcDsScienceDataServer <i>Libraries:</i> DsCI, DsSh <i>Classes:</i> DsCIRequest, DsCICommand, DsCIESDTReferenceCollector	The EcDsScienceDataServer returns a list of granule URs as a result of a query from the <b>EcCsMtMGateway</b> .

**Table 4.1-4. SDSRV CSCI Process Interface Events (10 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return Inventory Granule Metadata	Once per granule inspect	<i>Process:</i> EcCsMtMGateway <i>Class:</i> EcCsMtMECSSearchProxy, EcCsMtMSearchImp, EcCsMtMSdsrvMgr, EcCsMtMDaServerMgr	<i>Process:</i> EcDsScienceDataServer <i>Libraries:</i> DsCl, DsSh <i>Classes:</i> DsCIESDTReference, DsCIESDTReferenceCollector	The EcDsScienceDataServer returns granule metadata information as results of receiving a granule inspect request from the <b>EcCsMtMGateway</b> .
Get Info for EDN	One per request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCl <i>Class:</i> DsCIESDTReference	<i>Process:</i> EcCsEmailParser <i>Class:</i> EcCsEmailParser	The <b>EcCsEmailParser</b> sends requests to the EcDsScienceDataServer for the Expedited Data Set Notification (EDN) information and sends messages to users.
Send Search	Per client request	<i>Processes:</i> EcDsScienceDataServer <i>Library:</i> DsCl <i>Class:</i> DsCIESDTReferenceCollector	<i>Process:</i> EcCsMtMGateway <i>Library:</i> DsXSsStaticClientLibs (macro) <i>Classes:</i> See note at the end of this table.	The <b>EcCsMtMGateway</b> sends inventory search requests, constructed from the qualifying metadata information in the SIPS request, to the EcDsScienceDataServer via a remote procedure call.
Send Event ID	One per event	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeEventCustomizer	<i>Process:</i> EcSbSubServer <i>Library:</i> EcSbSrSh <i>Class:</i> EcSbEvent	The <b>EcSbSubServer</b> sends Event IDs to the EcDsScienceDataServer when ESDTs are installed or when ESDTs are updated by adding additional events.

**Table 4.1-4. SDSRV CSCI Process Interface Events (11 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	One per server	<i>Process:</i> EcDsScienceDataServer	<i>Process:</i> EcCsIdNameServer <i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodec <i>Classes:</i> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDsScienceDataServer retrieves server location information from the <b>EcCsIdNameServer</b> .
Return Configuration Parameters	One set per request	<i>Process:</i> EcDsScienceDataServer	<i>Process:</i> EcCsRegistry <i>Library:</i> EcCsRegistry <i>Class:</i> EcRgRegistryServer_C	The <b>EcCsRegistry</b> returns the attribute-value pairs (configuration parameters) to the EcDsScienceDataServer upon request.
Return Dist. Media Options	One set of media types per request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeServiceCustomizer	<i>Process:</i> EcCsRegistry <i>Library:</i> EcCsRegistry <i>Class:</i> EcRgRegistryServer_C	The <b>EcCsRegistry</b> returns the attribute-value pairs (configuration parameters) to the EcDsScienceDataServer upon request.
Search Inventory	One per service request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIQuery, DsCIESDTReferenceCollector	<i>Process:</i> EcDmV0ToGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwInventoryRequest	The <b>EcDmV0ToEcsGateway</b> sends requests to the EcDsScienceDataServer to search the inventory (archives).
Request Inspect	One per price estimate request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwPriceEstimateRequest	The <b>EcDmV0ToEcsGateway</b> sends a request for an inspection of granule metadata to the EcDsScienceDataServer in support of a price estimate request.

**Table 4.1-4. SDSRV CSCI Process Interface Events (12 of 12)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return Inventory Granule Metadata	One per service request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwInventoryRequest	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCl <i>Classes:</i> DsClQuery, DsCIESDTReferenceCollector	The EcDsScienceDataServer returns ESDT Universal References (URs) for the requested granules to the <b>EcDmV0ToEcsGateway</b> .
Return Granule URs	One per price estimate request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwPriceEstimateRequest	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCl <i>Classes:</i> DsCIESDTReferenceCollector, DsClRequest	The EcDsScienceDataServer returns the URs for the granules to the <b>EcDmV0ToEcsGateway</b> to inspect the metadata in support of a price estimate request.
Send ESDT Installation Information	One per new ESDT installation	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCISchemaRequest	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeDataDictController	The EcDsScienceDataServer sends ESDT installation information, to the <b>EcDmDictServer</b> , whenever a new ESDT is installed. This data consists of Inventory and Collection level metadata.
Replace ESDT Installation Information	One per ESDT update	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCISchemaRequest	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeDataDictController	The EcDsScienceDataServer sends updated ESDT information to the <b>EcDmDictServer</b> whenever an ESDT is updated. This data consists of updated Inventory and Collection level metadata.

**Note: DsXssStaticClientLibs is a macro defined in the COMMON components. It defines DSS library includes for the Cl, Sh, Ge, Sr, De2 and Gl libraries. The important classes used from this macro (by library include) are:**

- **Cl** – DsClCommand, DsClDescriptor, DsCIESDTAddRequest, DsCIESDTReference, DsCIESDTReferenceCollector, DsCIESDTReferenceVector, DsClInsertCommand, DsClQuery and DsClRequest
- **Sh** – DsShByteBuffer, DsShConnectionProxy, DsSh DescriptorProxy, DsShESDTUR, DsShError, DsShErrorDetails, DsShMutex, DsShMutexLock, DsShRequest, DsShRequestProxy and DsShThread

- **Gl** – GlParameterList, GlParameter, GlLongP, GlGPolygonP, GlDoubleP, GlBinaryP, GlRectangleP, GlStringP and GlTimeP

#### 4.1.1.6 Science Data Server Data Stores

Table 4.1-5 provides a description of the data stores for the SDSRV CSCI, and the conceptual model of the data store. The physical model for the SDSRV data stores can be found in the Science Data Server Database Design and Schema Specifications for the EMD Project (CDRL 311).

**Table 4.1-5. SDSRV CSCI Data Stores**

Data Store	Type	Description
EMD Inventory	Database	<p>The EMD Inventory (archives) contains the metadata describing the earth science data for the Earth Science Data Types at a specific DAAC. The metadata describes:</p> <ul style="list-style-type: none"> <li>• Collection level information</li> <li>• Browse data</li> <li>• Science data (as granules)</li> <li>• Quality Assessments</li> <li>• Algorithm Packages</li> <li>• Delivered Algorithm Packages</li> <li>• Production History</li> </ul> <p>The EMD catalog also contains systems data for the dynamic configuration of the EcDsScienceDataServer.</p> <p>The EMD catalog also contains implementation of the “EMD Data Model” for Attribute Validity checking.</p> <p>The EMD catalog also contains system data for ESDT Configuration.</p>

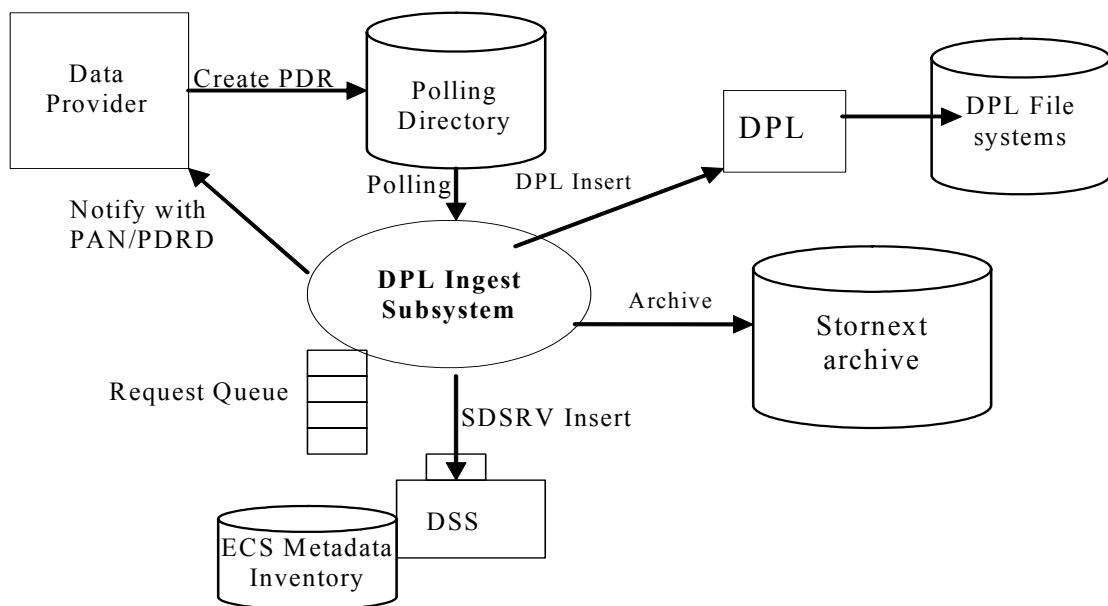
#### **4.1.2 Data Distribution Software Description (REMOVED)**

## 4.2 DPL Ingest Subsystem Overview

The Data Pool Ingest service will handle the SIPS ingest interface<sup>1</sup>, S4P, cross-DAAC ingest, EDOS ingest, ASTER Ingest and Polling without Delivery Record specifically for EMOS. Unlike the classic INGST subsystem, the Data Pool Ingest service will insert the ingested data into the Data Pool, in addition to inserting the ingested data into the archive.

### DPL Ingest Subsystem Context

Figure 4.2-1 is the DPL Ingest context diagram. The diagram provides an illustration of the Data Pool Ingest and archiving steps. Table 4.2-1 provides descriptions of the interface events shown in the DPL Ingest Subsystem context diagram.



**Figure 4.2-1. DPL Ingest Subsystem Context Diagram**

<sup>1</sup> EOSDIS Document 423-41-57 Interface Control Document between the ECS and the SIPS Volume 0 revision H

**Table 4.2-1. DPL Ingest Subsystem Interface Events**

Event	Interface Event Description
Create PDR	SIPS providers place their data and PDR files into a polling directory. The directory can be local, e.g., accessible via a mount point; or remote, i.e., accessible via FTP.
Polling	The DPL Ingest Service polls these directories as configured by the DAAC.
Request Queue	The DPL Ingest Service queues ingest requests for validation and processing. The DPL Ingest Service queues all PDR that it finds. To decide which validated PDR will be processed next, it checks available resources and DAAC configured priorities.
DPL Insert	The granule files are copied into the Data Pool SAN, using hidden directories for that purpose unless the DAAC requested that the data be published on insert.
Archive	The DPL Ingest Service then copies the granules from the hidden Data Pool directories into the StorNext archive.
SDSRV Insert	The granule metadata is inserted into the ECS inventory.
SSS notification	The DPL Ingest service places a record for the Spatial subscription server to decide whether any subscription should fire based on the granule insert.
Notify with PAN or PDRD	The notification could be immediate via PDRD if PDR validation failed, or later on via a short or long PAN. The provider is notified of the outcome.
Queue for Publication	The DPL Ingest service places a record for the granule to be published if its collection is marked for publication. The DPAD will then perform the publication.

### DPL Ingest Subsystem Structure

The DPL Ingest Subsystem consists of four CSCIs: the Polling Service, the Processing Service, the Notification Service, and the DPL Ingest GUI. The Polling Service is responsible for the provision of work to the service via transferring Product Delivery Records (PDRs) into the system and registering them. The Processing Service picks up registered PDRs and attempt to ingest the inventory they describe into the Data Pool and the Archive, performing any additional processing required for specific inventory. The Processing component will status a particular PDR on completion of various steps during processing. The Notification Service will detect registered PDRs that have reached a terminal state within the system and notify the provider associated with that PDR of that state. Terminal states are Successful, Partially Failed, Failed, Cancelled, and Partially Cancelled. Terminal states are conveyed to the provider by means of a Product Acceptance Notification (PAN) or Product Delivery Discrepancy Report (PDRD). The DPL Ingest GUI is used to monitor and control the operations of the DPL Ingest Service. In addition, the DPL Ingest GUI is used to respond to Operator Intervention Requests generated by the DPL Processing Service.

### Use of COTS in the DPL Ingest Subsystem

- RogueWave's Tools.h++

The Tools.h++ class libraries are used by the DPL Ingest Service to provide basic functions and objects such as strings and collections. These libraries must be installed with the DPL Ingest software for any of the DPL Ingest Service processes to run.

- Sybase Open Client / CT\_LIB  
The Sybase Open Client provides access between DPL Ingest Service custom code and the Sybase SQL Server DBMS.
- Sybase Server  
The Sybase SQL server provides access for DPL Ingest Service to insert, update and delete DPL Ingest Requests, DPL Ingest configurations, and Operator Interventions. The Sybase SQL Server must be running during operations for the DPL Ingest Service to process DPL Ingest Requests.
- CCS Middleware Client  
This product provides the communications between DPL Ingest and other subsystems. CCS Middleware can reside on one or both sides of the interface and must be installed on the platform where the DPL Ingest resides. Although the CCS Middleware Client is part of the CSS, this COTS product must be installed on the platform where the DPL Ingest software resides for DPL Ingest to run in the ECS operational and test environments.
- UNIX Network Services  
DNS, NFS, E-mail, FTP, TCP/IP and the other Unix services provided are obtained from the CSS.

## **4.2.1 DPL Ingest Computer Software Configuration Item Description**

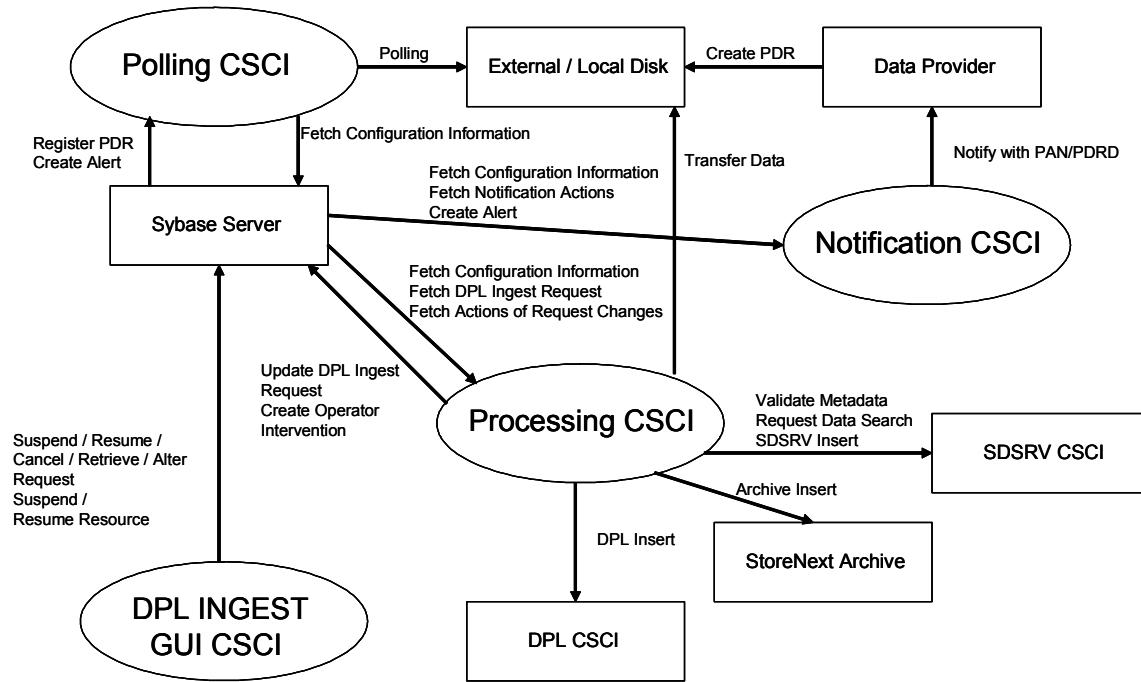
### **4.2.1.1 DPL Ingest Service CSCI Functional Overview**

The DPL Ingest Subsystem consists of four CSCIs: the Polling Service, the Processing Service, the Notification Service, and the DPL Ingest GUI. The Processing Service executes as a process and interacts with the following CSCIs: INGST Database, the Polling Service, the Notification Service, the Science Data Server (SDSRV), and the Data Pool System (DPL). The Polling Service transfers Product Delivery Records (PDRs) into the system and registers them to the INGST Database, and these requests are all validated by the Processing Service. If the request is valid, then Processing Service attempts to ingest the inventory they describe into the Data Pool and Archive, performing any additional processing required for specific inventory. The Processing Service updates the status a particular PDR on completion of various steps during processing. For invalid request, an Operator Intervention is generated. DAAC OPS personnel can use the DPL Ingest GUI to correct and resubmit the request. In response to an intervention, the Operator can also generate an email message, which is sent to the user by the Notification Service CSCI. The Processing Service CSCI also generates an alert when it detects internal or external resource failure, and Notification Service CSCI sends an email to a pre-configured email address. In the meanwhile, the Processing Service CSCI halts dispatching of the requests that are utilizing those failed resources.

### **4.3.1.2 DPL Ingest Service CSCI Context**

Figures 4.2-2 is the DPL Ingest Service CSCI context diagrams. The diagrams show the events sent to the DPL Ingest Service CSCIs and the events the DPL Ingest CSCIs send to other CSCIs.

Table 4.2-2 provides descriptions of the interface events shown in the DPL Ingest Service CSCI context diagram.



**Figure 4.2-2. DPL Ingest CSCI Context Diagram**

**Table 4.2-2. DPL Ingest CSCI Interface Events (1 of 2)**

Event	Interface Event Description
Create PDR	SIPS providers place their data and PDR files into a polling directory which will be polled by the Polling CSCI. The directory can be local, e.g., accessible via a mount point; or remote, i.e., accessible via FTP.
Polling	The Polling CSCI polls PDRs from directories ( <b>External/Local Disk</b> ) by <b>Data Provider</b> as configured by the DAAC.
Register PDR	The Polling CSCI queues ingest requests for validation and processing into the <b>Sybase Server (INGST Database)</b> . The Processing CSCI later queues all PDR that it finds. To decide which validated PDR will be processed next, it checks available resources and DAAC configured priorities.
DPL Insert	The Processing CSCI copies the granule files into the <b>Data Pool SAN</b> , using hidden directories for that purpose unless the DAAC requested that the data be published on insert.
Archive Insert	The Processing CSCI copies the granule files in to the StoreNext archive.
SDSRV Insert	The Processing CSCI also inserts the granules into the <b>SDSRV</b> inventory.
Notify with PAN or PDRD	The Notification CSCI sends notification to the <b>Data Provider</b> , it could be immediate via PDRD if PDR validation failed, or later on via a short or long PAN.

**Table 4.2-2. DPL Ingest CSCI Interface Events (2 of 2)**

Event	Interface Event Description
Create Alert	The Polling CSCI, Processing CSCI and Notification CSCI creates an alert for resource failures and stores the alert in the <b>Sybase Server (INGST Database)</b> .
Fetch Config Info	The Polling CSCI, Processing CSCI and Notification CSCI retrieves the configuration information from <b>Sybase Server (INGST Database)</b> .
Update DPL Ingest Request	The Processing CSCI updates DPL Ingest request in the <b>Sybase Server (INGST Database)</b> .
Create Operator Intervention	The Polling CSCI, Processing CSCI creates new Operator Intervention for request failures in the <b>Sybase Server (INGST Database)</b> .
Fetch DPL Ingest Request	The Processing CSCI retrieves information associated with a DPL Ingest request from the <b>Sybase Server (INGST Database)</b> .
Fetch Actions of Request Changes	The Processing CSCI retrieves actions regarding request changes, such as, request priority change, cancel request, suspend request, and update request parameters from the <b>Sybase Server (INGST Database)</b> .
Validate Metadata	The Processing CSCI populates the metadata files and sends them to the <b>SDSRV CSCI</b> for validation.
Request Data Search	The Processing CSCI requests a search, by the <b>SDSRV CSCI</b> , for the granule corresponding to a particular ESDT short name and version, which has a particular local granule id.
Fetch Notification Actions	The Notification CSCI retrieves actions regarding request notifications from the <b>Sybase Server (INGST Database)</b> .
Suspend/Resume/Cancel/Alter/Retrieve Request	The DPL Ingest GUI CSCI suspends, resumes, cancels, alters and retrieves requests from the <b>Sybase Server (INGST Database)</b> .
Suspend/Resume Resource	The DPL Ingest GUI CSCI suspends or resumes dispatching to all or selected resources in the <b>Sybase Server (INGST Database)</b> .
Transfer Data	The Processing CSCI transfers data files from the <b>External/Local Disk</b> specified in PDR.

#### 4.2.1.3 DPL Ingest Architecture

The Polling Ingest Interface (EcDlInPollingService) polls accessible file system locations to detect data to be ingested. This process submits a Product Delivery Record (PDR). The Cross-Mode Ingest Interface (EcInEmailGWServer) receives distribution notifications, via e-mail, and stores them as files in a location, which is polled. This process detects the notification files and creates Delivery Record files, which are put in a polling directory and detected by the Polling Ingest Interface.

The Polling Ingest Interface queues ingest requests for validation and processing into the Sybase Server (INGST database). The Processing Interface queues all PDR that it finds, to decide which validated PDR will be processed next, it checks available resources and DAAC configured priorities. The Processing Interface validates metadata through SDSRV and inserts the granules into SDSRV inventory. The Processing Interface copies the granule files into Data Pool SAN, using hidden directories for that purpose unless the DAAC requested that the data be published on insert. The processing Interface copies the granule files into the StorNext archive.

Figure 4.2-3 is the DPL Ingest CSCI architecture diagram. The diagram shows the events sent to the DPL Ingest CSCI processes and the events the DPL Ingest CSCI processes send to other processes.

**Note: System startup and shutdown** - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).

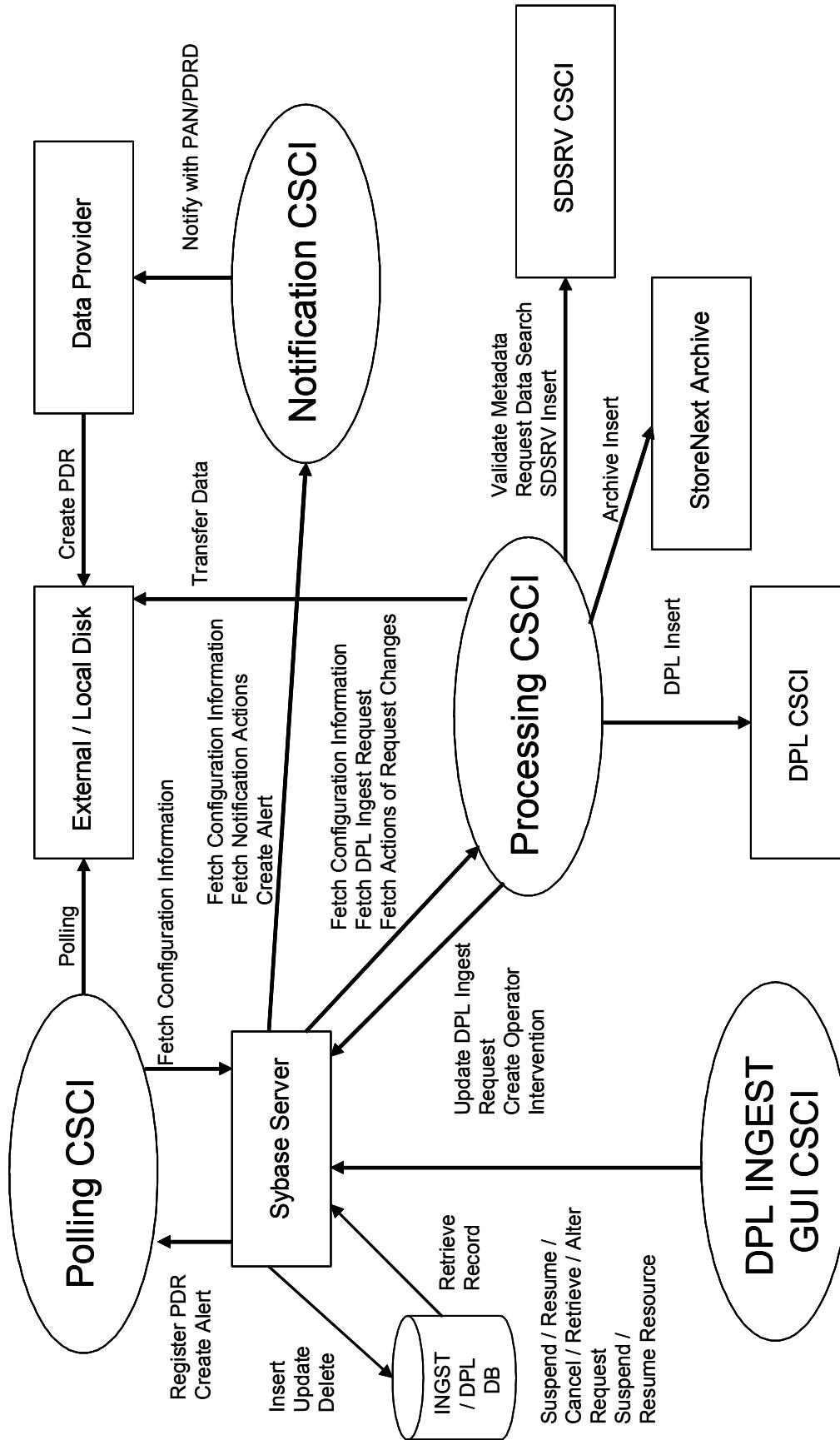


Figure 4.2-3. DPL Ingest CSCl Architecture Diagram

#### 4.2.1.4 DPL Ingest Process Descriptions

Table 4.2-3 provides the descriptions of the processes shown in the DPL Ingest CSCI architecture diagram (Figure 4.2-3).

**Table 4.2-3. DPL Ingest CSCI Processes**

Process	Type	Hardware CI	Source	Functionality
EcDIInPolling Service	Server	DPLHW	Developed	<ul style="list-style-type: none"> <li>• Detect new Product Delivery Records (PDRs) and transfer them into system.</li> <li>• Creates a unique identifier for the request</li> <li>• Register request</li> </ul>
EcDIInGui	GUI	INTHW	Developed	<p>Provides Maintenance and Operations (M&amp;O) personnel the capability, via GUI Interface,</p> <ul style="list-style-type: none"> <li>• To modify ingest configuration parameters.</li> <li>• To monitor the status of ongoing ingest requests, to suspend, resume, cancel, alter or retrieve DPL ingest requests.</li> <li>• To suspend or resume resource.</li> </ul>
EcDIInProcessingService	Server	DPLHW	Developed	<ul style="list-style-type: none"> <li>• Ingests granules associated with ingest requests (PDRS) into the Datapool and archive.</li> <li>• Registers granule information with the science data server</li> <li>• </li> <li>• Manages the DPL ingest request traffic and the processing of the DPL ingest requests, and</li> <li>• Provides the capability to process multiple ingest requests concurrently by placing the request in a queue</li> <li>• In the event of a failure, the EcDIInProcessingService process restores on-going requests from the Ingest database</li> </ul>
EcDIInNotificationService	Server	DPLHW	Developed	<ul style="list-style-type: none"> <li>• Send the end-user Notification, either Product Acceptance (PAN) or Product Delivery Discrepancy Report (PDRD), on completing a ingest request.</li> </ul>
Sybase	Server	ACMHW	COTS	<ul style="list-style-type: none"> <li>• Stores and provides access to the DPL Ingest Service internal data. In particular, the database stores the Ingest operations databases – DPL Ingest History Logs and the DPL Ingest request checkpoint state, and template information. See Section 4.2.1.6 DPL Ingest Data Stores.</li> </ul>

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

#### 4.2.1.5 DPL Ingest Process Interface Descriptions

Table 4.2-4 provides descriptions of the interface events shown in the DPL Ingest CSCI Architecture diagram.

**Table 4.2-4. DPL Ingest CSCI Process Interface Events (1 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Create PDR	One per request	Directories on remote or local disk	External Data Provider	SIPS providers place their data and PDR files into a polling directory which will be polled by the EcDlInPollingService. The directory can be local, e.g., accessible via a mount point; or remote, i.e., accessible via FTP.
Polling	One per request	Directories on remote or local disk	<i>Process:</i> EcDlInPollingService <i>Class:</i> DplnPoller	The EcDlInPollingService polls PDRs from directories (External/Local Disk) by Data Provider as configured by the DAAC.
Register PDR	One per request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInPollingService <i>Class:</i> DplnPollingLocation	The EcDlInPollingService queues ingest requests for validation and processing into the Sybase Server (INGST database). The EcDlInProcessingService later queues all PDR that it finds. To decide which validated PDR will be processed next, it checks available resources and DAAC configured priorities.
DPL Insert	One per request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplnInternalFtpTransferQAction	The EcDlInProcessingService copies the granule files into the Data Pool SAN, using hidden directories for that purpose unless the DAAC requested that the data be published on insert.
Archive Insert	One per request	<i>Process:</i> StoreNext copy	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplnInternalFtpTransferQAction	The EcDlInProcessingService copies the granule files into the StoreNext archive file system

**Table 4.2-4. DPL Ingest CSCI Process Interface Events (2 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
SDSRV Insert	One per request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Class:</i> DplnSdsrvInsertServiceQAction DsCIRequest	<i>Process:</i> EcDlInProcessingService <i>Service</i> <i>Class:</i> DplnNotifyEmailAction DplnNotifyFtpAction	The EcDlInProcessingService also inserts the granules into the SDSRV inventory.
Notify with PAN or PDRD	One per email notification request	<i>Process:</i> Sendmail (COTS) Ftp daemon (COTS)	<i>Process:</i> EcDlInNotificationService <i>Service</i> <i>Class:</i> DplnNotifyEmailAction DplnNotifyFtpAction	The EcDlInNotificationService sends notification to the Data Provider, it could be immediate via PDRD if PDR validation failed, or later on via a short or long PAN.
Create Alert	One per request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInPollingService EcDlInProcessingService EcDlInNotificationService <i>Service</i> <i>Class:</i> DpCoAlert	The EcDlInPollingService, EcDlInProcessingService and EcDlInNotificationService creates an alert for resource failures and stores the alert in the Sybase Server (INGST Database).
Fetch Config Info	One per startup/ One per configurable interval	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInPollingService EcDlInProcessingService EcDlInNotificationService <i>Service</i> <i>Class:</i> DplnNotifyDatabase	The EcDlInPollingService, EcDlInProcessingService and EcDlInNotificationService retrieve the configuration information from Sybase Server (INGST Database).
Update DPL Ingest Request	One per request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInProcessingService <i>Service</i> <i>Class:</i> DplnProcessingDbInterface	The EcDlInProcessingService updates DPL Ingest request in the Sybase Server (INGST Database).
Create Operator Intervention	One per request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInProcessingService <i>Service</i> <i>Class:</i> DplnProcessingDbInterface	The EcDlInProcessingService creates new Operator Intervention for request failures in the Sybase Server (INGST Database).

**Table 4.2-4. DPL Ingest CSCI Process Interface Events (3 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Fetch DPL Ingest Request	One per request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplnProcessingDbInterface	The EcDlInProcessingService retrieves information associated with a DPL Ingest request from the Sybase Server (INGST Database).
Fetch Actions of Request Changes	One per configurable interval	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplnProcessingDbInterface	The EcDlInProcessingService retrieves actions regarding request changes, such as, request priority change, cancel request, suspend request, and update request parameters from the Sybase Server (INGST Database).
Validate Metadata	One per metadata validation	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCl <i>Class:</i> DsClDescriptor	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplnGranuleScheduler	The EcDlInProcessingService populates the metadata files and sends them to the SDSRV CSCI for validation.
Request Data Search	One per granule pointer in linkage file	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplnGranuleScheduler	The EcDlInProcessingService requests a search, by the SDSRV CSCI, for the granule corresponding to a particular ESDT short name and version, which has a particular local granule id.
Fetch Notification Actions	One per configurable interval	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDlInNotificationService <i>Class:</i> DplnNotifyDatabase	The EcDlInNotificationService retrieves actions regarding request notifications from the Sybase Server (INGST Database).
Suspend/Resume/Cancel/Alter/Retrieive Request	One per click	<i>Process:</i> Sybase Server (COTS)	Script: DPL Ingest GUI script	The DPL Ingest GUI scripts send suspend, resume, cancel, alter and retrieve request command to the Sybase Server (INGST Database).
Suspend/Resume Resource	One per click	<i>Process:</i> Sybase Server (COTS)	Script: DPL Ingest GUI script	The DPL Ingest GUI scripts send suspend or resume resource command to the Sybase Server (INGST Database).

**Table 4.2-4. DPL Ingest CSCI Process Interface Events (4 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Transfer Data	One per science data file activity	<i>Process:</i> Ftpd (COTS)	<i>Process:</i> EcDlInProcessingService <i>Class:</i> DplInternalFtpTransferQAction	The EcDlInProcessingService transfers data files from the External/Local Disk specified in PDR.

#### **4.2.1.6 Ingest Data Stores**

The DPL Ingest CSCI uses the COTS product Sybase to store related DPL Ingest Information. Table 4.2-5 provides descriptions of the data stores.

**Table 4.2-5. DPL Ingest CSCI Data Stores**

Data Store	Type	Description
INGST Database	Sybase	INGST Database is designed to store the persistent information of user request, processing configuration, request aging configuration, and request cleanup configuration.
DPL Database	Sybase	The Data Pool (DPL) database implements the large majority of the persistent data requirements for the DPL subsystem which supports large online cache of important ECS data at each DAAC and avoids tape access to ECS archive.

### 4.3 Client Subsystem Overview

The Client Subsystem (CLS) is a set of CSCIs and processes that provide EMD end-user services.

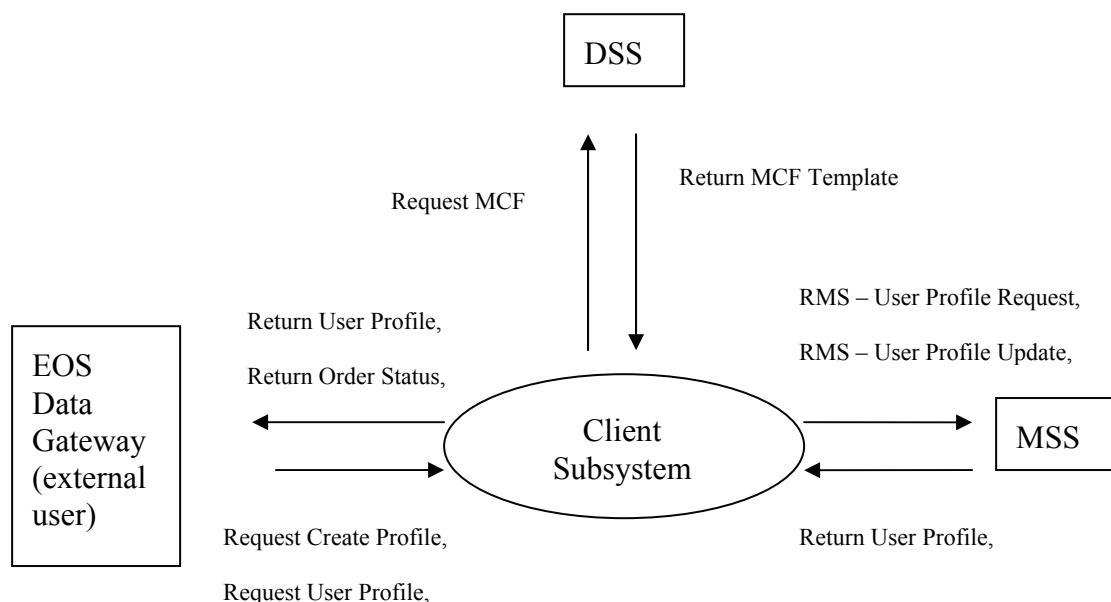
These services include allowing users to:

- Create, retrieve and update ECS user profiles
- View HDF formatted files

In addition, the workstations operating within an ECS CLS contains infrastructure support software as part of the CSS and platform support software.

#### Client Subsystem Context

Figure 4.3-1 is the Client Subsystem context diagrams. The diagrams show the events sent to the CLS and the events the CLS sends to other CSMS subsystems.



**Figure 4.3-1. Client Subsystem Context Diagram**

The EOS Data Gateway, the Version 0 Client (Web version) performs the data searches and retrievals. The EOS Data Gateway is treated as an external entity since the design and design documentation is controlled under another contract. Documentation on the V0 Web Client (EOS

Data Gateway) design can be accessed through the Universal Resource Locator <http://edg.larc.nasa.gov/~imswww/pub/imswelcome/imswwwsites.html>.

Table 4.3-1 provides descriptions of the interface events shown in the Client Subsystem context diagram.

**Table 4.3-1. Client Subsystem Interface Events (1 of 2)**

Event	Interface Event Description
Enter HDF File Name	This is a file name for a Hierarchical Data Format (HDF) file. The <b>user</b> opens the file to see the data in the file.
Request Login	A user name and password for the <b>user</b> are provided for access to the ECS. The user name and password are sent to the CSS (via a request for communications support).
Request Management Services	The <b>MSS</b> provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services includes: <ul style="list-style-type: none"> <li>• <b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).</li> <li>• <b>Create Profile Request</b> - The <b>MSS</b> receives user information for becoming a registered user of the ECS from the CLS. The MSS sends a response to the user when the request is received.</li> <li>• <b>User Profile Request</b> - The <b>MSS</b> provides requesting subsystems with User Profile parameters such as e-mail address and shipping address upon request by authorized users to support their processing activities.</li> <li>• <b>User Profile Updates</b> - The <b>MSS</b> receives user profile parameter updates from a user and makes the updates in the user profile database.</li> <li>• <b>Password Change Request</b> - The CLS sends requests on behalf of ECS users to the <b>MSS</b> to change users' authenticators in the MSS database.</li> </ul>
Return User Profile	The user profile is returned from the <b>MSS</b> to the CLS to be returned to the external user via the EOS Data Gateway.
Return Order Status	The CLS receives an order id and status for the requested ECS product from the <b>MSS</b> and returns the order id and status to the ECS or V0 IMS (via the EDG) user (to track the order).
Request Create Profile	Users submit a request to be a registered user of the ECS. Registered users can be given special privileges not awarded to guests, such as the capability to order data on a media at a cost. The user request is sent through the <b>EOS Data Gateway (EDG)</b> , which creates both an EDG and an ECS User Profile. The user enters his or her addresses (user, shipping, billing and e-mail) and other important information. This profile information is used to establish an EDG profile and is forwarded through the CLS to the MSS. The MSS creates the ECS profile.

**Table 4.3-1. Client Subsystem Interface Events (2 of 2)**

Event	Interface Event Description
Request User Profile	A User Profile Request is a search for a User Profile from the user via the <b>workbench</b> or <b>EDG</b> . In response, the CLS receives the user profile, which contains information about a user that must be maintained. This includes, but is not limited to, mailing, billing, and shipping addresses, phone number, electronic mail address, project account number and project organization.
Request User Profile Updates	The user can update their User Profile information through the <b>EOS Data Gateway</b> . This includes their addresses (user, shipping, billing, and e-mail) and other important information. This updated profile information (profile2.odl) is forwarded through the CLS to the MSS. The EOS Data Gateway uses the profile2.odl file rather than a live interface with the CLS as its source of user information.
Request Password Change	The user can request a change of his/her EDG User Profile passwords through the <b>EOS Data Gateway</b> . The user enters his current and new passwords. This information is used to change the EDG password and the passwords (old and new) are forwarded through the CLS to the MSS. The MSS changes the ECS password.
Request MCF	Requests the Metadata Control File (MCF) for a particular ESDT
Import Location Information	The CLS requests server location information from the <b>CSS</b> (CCS NameServer).

## Client Subsystem Structure

The Client Subsystem is two CSCIs:

- The Desktop (DESKT) CSCI provides the User Profile Gateway server to communicate with the MSS User Registration Server for creating new user accounts, obtaining user profile information to update user information.
- The Science Software Integration and Test (SSIT) CSCI was migrated from the Data Processing Subsystem (DPS) to CLS when the former was discontinued. It has two tools: Get MCF and Hdiff

In addition, the User Profile Gateway Server is used to create, retrieve and update user profiles for ECS users via the EOS Data Gateway. The CLS is required to support Linux hosts.

The Interface Server is documented in 920-TDx-001 ('x' is denoted for the DAAC) which provides descriptions of the Interface Server HWCI. The Workbench software executed on these hosts enables user access to the ECS data and services.

## Error Handling and processing

EcUtStatus is a class used throughout the ECS custom code for general error reporting. It is almost always used as a return value for functions and allows detailed error codes to be passed back up function stacks.

The DESKT CSCI consists of the User Profile Gateway. The User Profile Gateway uses EcUtStatus and exceptions. Functions can return an EcUtStatus object, indicating success or

failure or throw an exception. The User Profile Gateway does not have any special error processing.

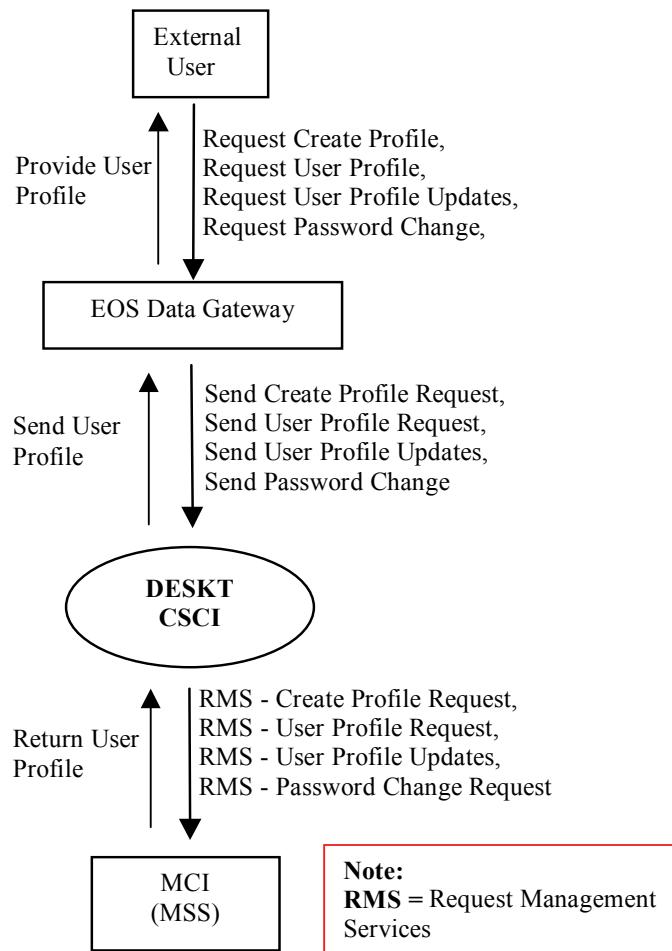
### 4.3.1 Desktop Software Description

#### 4.3.1.1 Desktop Functional Overview

The DESKT CSCI provides the User Profile Gateway server to communicate with the MSS User Registration Server for obtaining user profile information to authenticate users or update user information.

#### 4.3.1.2 Desktop Context

Figure 4.3-2 is the DESKT CSCI context diagram. The diagram shows the events sent to the DESKT CSCI and events the DESKT CSCI sends to other CSCIs. Table 4.3-2 provides descriptions of the interface events shown in the DESKT CSCI context diagram.



**Figure 4.3-2. DESKT CSCI Context Diagram**

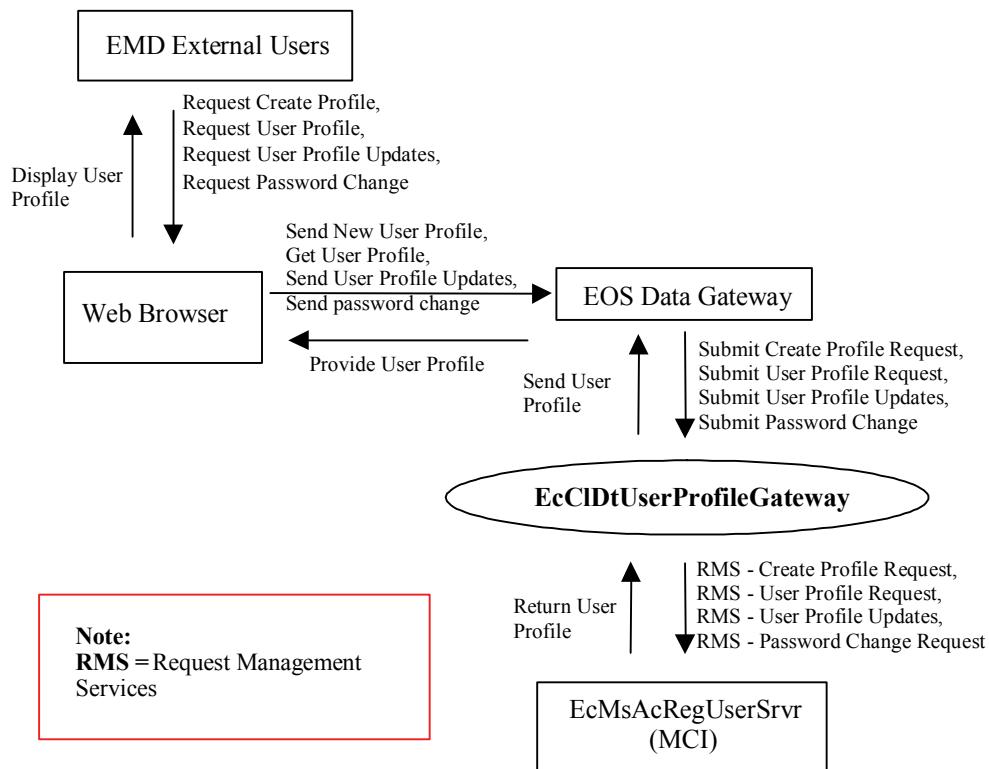
**Table 4.3-2. DESKT CSCI Interface Events**

Event	Interface Event Description
Request Create Profile	A <b>user</b> submits a request to be a registered user of the ECS to the EOS Data Gateway (to forward to the MCI) to create a user profile. The user profile includes, but is not limited to, mailing, billing and shipping contact information.
Request User Profile	The <b>user</b> sends a request to the EOS Data Gateway (to the MCI) for retrieval of a User Profile. The user profile is the important information about a user that must be maintained. This includes, but is not limited to, mailing, billing, and shipping addresses, phone number, and electronic mail address.
Request User Profile Updates	<b>Users</b> can update their User Profile information through the EOS Data Gateway (EDG). This includes their addresses (user, shipping, billing, and e-mail) and other information. This file gets forwarded to the DESKT CSCI and the DESKT CSCI forwards the file to the MCI.
Request Password Change	The <b>user</b> can request a change or reset of his/her EDG/ECS User Profile passwords through the EOS DATA Gateway. The user enters his userid, current and new passwords. The userid and passwords are converted to authenticators. This information is used to change the EDG password and the authenticators (old and new) are forwarded through the CLS to the MSS. The MSS changes the ECS authenticator. The authenticator is an encrypted version of the userid and password. It is encrypted so it can be passed over a socket without threat of being stolen.
Send Create Profile Request	The <b>EOS Data Gateway</b> sends a request to the DESKT CSCI to register the user in the ECS.
Send User Profile Request	The <b>EOS Data Gateway</b> sends the user profile request to the DESKT CSCI for processing.
Send User Profile Updates	The <b>EOS Data Gateway</b> sends the user profile updates to the DESKT CSCI for processing.
Send Password Change	The <b>EOS Data Gateway</b> sends a request to the DESKT CSCI to change or reset the user's password in the ECS for user access.
Request Management Services	The <b>MCI</b> provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: <ul style="list-style-type: none"> <li>• <b>Create Profile Request</b> – The MCI receives user information for becoming a registered user of the ECS from the DESKT CSCI. The MCI sends a response to the user when the request is received.</li> <li>• <b>User Profile Request</b> – The MCI provides requesting CSCIs with User Profile parameters such as e-mail address and shipping address upon request by authorized users to support their processing activities.</li> <li>• <b>User Profile Updates</b> – The MCI receives user profile parameter updates from a user and makes the updates in the user profile database.</li> <li>• <b>Password Change Request</b> – The DESKT CSCI sends requests on behalf of ECS users to the MCI to change or reset users' authenticators in the MSS database.</li> </ul>
Return User Profile	The MSS Sybase ASE returns the user profile to the DESKT CSCI, via the <b>MCI</b> , to forward back to the user via the EOS Data Gateway.
Send User Profile	The DESKT CSCI returns the user profile to the <b>EOS Data Gateway</b> .
Provide User Profile	The EOS Data Gateway provides the user profile to the external user.

#### 4.3.1.3 Desktop Architecture

Figure 4.3-3 is the DESKT CSCI architecture diagram. The diagram shows the events sent to the DESKT CSCI processes and the events the DESKT CSCI processes send to other processes.

The DESKT CSCI consists of one process. This process is the User Profile Gateway, a server that listens for calls on a socket. This process resides inside the DAAC on the INTHW server.



**Figure 4.3-3. DESKT CSCI Architecture Diagram**

#### 4.3.1.4 Desktop Process Description

Table 4.3-3 provides descriptions of the processes shown in the DESKT CSCI architecture diagram.

**Table 4.3-3. DESKT CSCI Processes**

Process	Type	COTS/ Developed	Functionality
EcCIDtUserProfileGateway	Server	Developed	<p>The User Profile Gateway is a retrieval and update engine for the user profile information. The server listens for calls on a socket.</p> <p>Interfaces:</p> <ul style="list-style-type: none"> <li>• User Profile Request: Request for a user profile given the user's authenticator.</li> <li>• Profile Update Request: Request to update the user's profile with new information.</li> </ul> <p>Server Supports:</p> <ul style="list-style-type: none"> <li>• Multiple concurrent requests.</li> </ul>

#### 4.3.1.5 Desktop Process Interface Descriptions

Table 4.3-4 provides descriptions of the interface events shown in the DESKT CSCI architecture diagram.

**Table 4.3-4. DESKT CSCI Process Interface Events (1 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Create Profile	Once per User	Process: Web Browser (COTS)	User	This is a request by a user to register (establish an account) in the ECS.
Request User Profile	One per User Login	Process: Web Browser (COTS)	User	The <b>user</b> sends a request to the EcMsAcRegUserSrvr using the ECS Authenticator from the EcCIDtUserProfileGateway or the user's ID.
Request User Profile Updates	One per User Profile update	Process: Web Browser (COTS)	User	<b>Users</b> can update their profiles via the Web Browser through the EOS Data Gateway.
Request Password Change	One per password change	Process: Web Browser (COTS)	User	<b>Users</b> can update or reset their passwords via the Web Browser through the EOS Data Gateway.
Send New User Profile	Once per User	Process: EOS Data Gateway	Process: Web Browser (COTS)	The Web Browser sends the User Registration request to the EOS Data Gateway for submittal to the ECS.
Get User Profile	One per User Profile	Process: EOS Data Gateway	Process: Web Browser (COTS)	The Web Browser sends the request to obtain a profile to the <b>EOS Data Gateway</b> .

**Table 4.3-4. DESKT CSCI Process Interface Events (2 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Send User Profile Updates	One set of parameters per request	<i>Process:</i> EOS Data Gateway	<i>Process:</i> Web Browser (COTS)	The Web Browser sends a request to update profile information to the <b>EOS Data Gateway</b> . This interface to the EcCIDtUserProfileGateway requests an update to the user's profile.
Send Password Change	Once per password change	<i>Process:</i> EOS Data Gateway	<i>Process:</i> Web Browser (COTS)	The Web Browser sends a password change or reset request to the <b>EOS Data Gateway</b> to be processed within the ECS.
Submit Create Profile Request	Once per User	<i>Process:</i> EcCIDtUserProfile Gateway <i>Class:</i> CIDtProfileServer	<i>Process:</i> EOS Data Gateway	The EOS Data Gateway submits the request to register a user to the EcCIDtUserProfileGateway and ultimately processed by the EcMsAcRegUserSrvr within the MSS.
Submit User Profile Request	One per user request	<i>Process:</i> EcCIDtUserProfile Gateway <i>Class:</i> CIDtProfileServer	<i>Process:</i> EOS Data Gateway	The <b>EOS Data Gateway</b> submits the request to obtain a profile to the EcCIDtUserProfileGateway and ultimately processed by the EcMsAcRegUserSrvr.
Submit User Profile Updates	One set of parameters per request	<i>Process:</i> EcCIDtUserProfile Gateway <i>Class:</i> CIDtProfileServer	<i>Process:</i> EOS Data Gateway	The <b>EOS Data Gateway</b> submits the request to update a profile to the EcCIDtUserProfileGateway and ultimately processed by the EcMsAcRegUserSrvr.
Submit Password Change	Once per password change	<i>Process:</i> EcCIDtUserProfile Gateway <i>Class:</i> CIDtProfileServer	<i>Process:</i> EOS Data Gateway	The <b>EOS Data Gateway</b> submits the request to change or reset a password to the EcCIDtUserProfileGateway, which is ultimately processed by the EcMsAcRegUserSrvr.

**Table 4.3-4. DESKT CSCI Process Interface Events (3 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services (RMS)	One per service request	N/A	N/A	The <b>EcMsAcRegUserSrvr</b> provides a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items listed below.
RMS (cont.)	One per user request	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Classes:</i> MsAcUsrProfile, RWPortal	<i>Process:</i> EcCIDtUserProfileGateway <i>Class:</i> CIDtProfileServer	<b>User Profile Request</b> – The <b>EcMsAcRegUserSrvr</b> provides requesting processes with User Profile parameters such as e-mail and shipping addresses to support their processing activities.
RMS (cont.)	One set of parameters per user request	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Classes:</i> MsAcUsrProfile, RWPortal	<i>Process:</i> EcCIDtUserProfileGateway <i>Class:</i> CIDtProfileServer	<b>User Profile Updates</b> – The <b>EcMsAcRegUserSrvr</b> provides requesting processes with access to User Profile parameters such as e-mail and shipping addresses to support the update of the parameters.
RMS (cont.)	One set of parameters per user request	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Classes:</i> MsAcUsrProfile, RWPortal	<i>Process:</i> EcCIDtUserProfileGateway <i>Class:</i> CIDtProfileServer	<b>Password Change Request</b> – The EcCIDtUserProfileGateway sends requests on behalf of ECS users to the <b>EcMsAcRegUserSrvr</b> to change or reset a user's password in the MSS database. The password is represented as an authenticator in the MSS database.

**Table 4.3-4. DESKT CSCI Process Interface Events (4 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return User Profile	One per user request	<i>Process:</i> EcCIDtUserProfile <i>Gateway</i> <i>Class:</i> CIDtProfileServer	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Classes:</i> MsAcUsrProfile, RWPortal	The <b>EcMsAcRegUserSrvr</b> returns the user profile to the EcCIDtUserProfileGateway.
Send User Profile	One per user request	<i>Process:</i> EOS Data Gateway	The EcCIDtUserProfile Gateway forwards the user profile to the EOS Data Gateway	The EcCIDtUserProfile Gateway forwards the user profile to the <b>EOS Data Gateway</b> .
Provide User Profile	One per user request	<i>Process:</i> Web Browser (COTS)	<i>Process:</i> EOS Data Gateway	The EOS Data Gateway returns the profile information to the Web Browser.
Display User Profile	One per user request	External Users	<i>Process:</i> Web Browser (COTS)	The Web Browser displays the user profile information to the external users.

#### 4.3.1.6 Desktop Data Stores

There are no data stores used by the DESKT CSCI.

#### 4.3.2 SSI&T Tools Description

Table 4.3-5 describes the SSI&T event descriptions.

**Table 4.3-5. SSI&T Tool Events**

Event	Event Frequency	Interface	Initiated By	Event Description
Hdiff hdiff	cmd line I/F and COTS binary	SSIT	Developed and COTS	Tools to compare binary and HDF files. The shell program PPAEcCIHDiff and is used to assist with the viewing and comparisons.
EcCIAcquireMCF EcCIGetMCF	cmd line I/F and Custom binart	SSIT	Developed	Tools to acquire an MCF from SDSRV (EcCIAcquireMCF) and associated binary, EcCIGetMCF

## **4.4 Data Management Subsystem Overview**

The Data Management Subsystem (DMS) provides catalog interoperability between the Version 0 (V0) Information Management System (IMS) and the ECS. The DMS provides this service by supplying gateway processes. There is a dedicated gateway process for each external catalog interoperability direction.

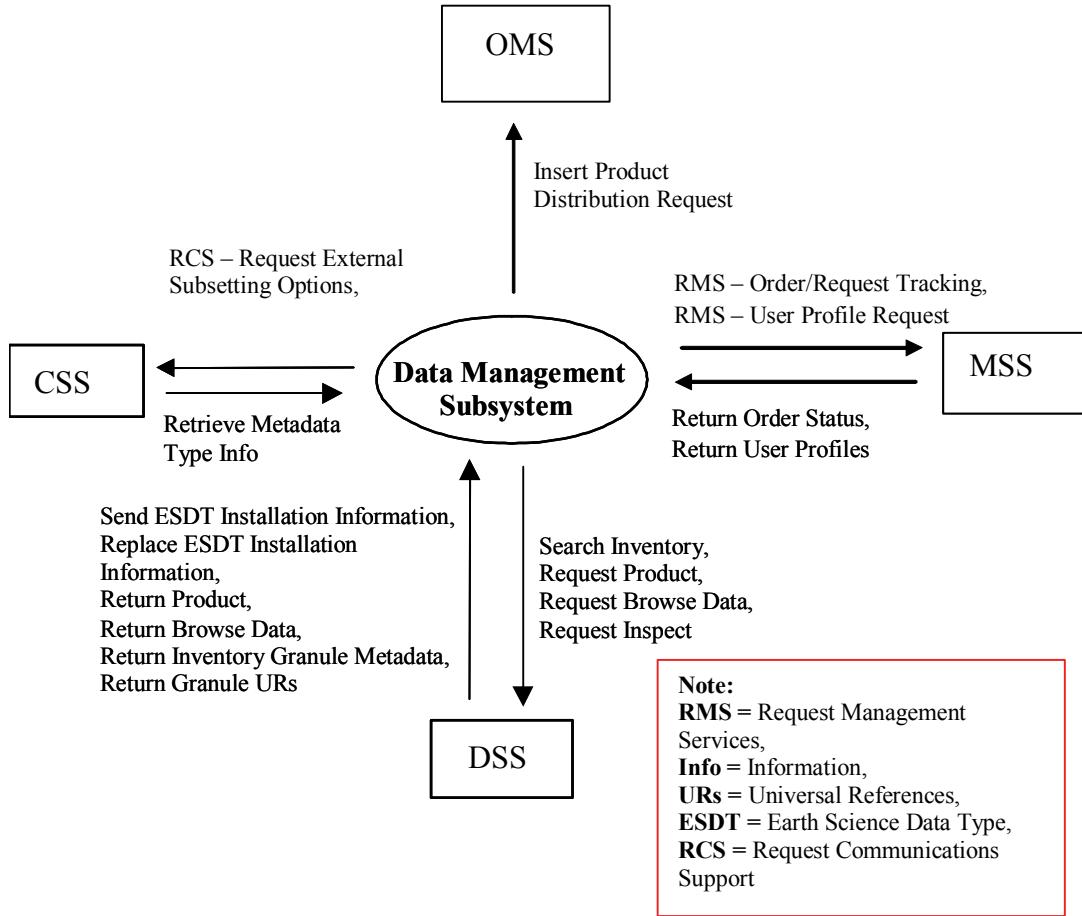
The DMS provides protocol translation via the dedicated gateways. The DMS provides content translation via a data dictionary of data collection information. This data dictionary contains collection level metadata in addition to attribute and keyword mappings between the ECS and the external catalog systems.

The V0 EOS Data Gateway (V0 Web Client) marshals the EMD interface to the V0-IMS. Documentation for the V0 Web Client can be found at the following Universal Resource Locator:

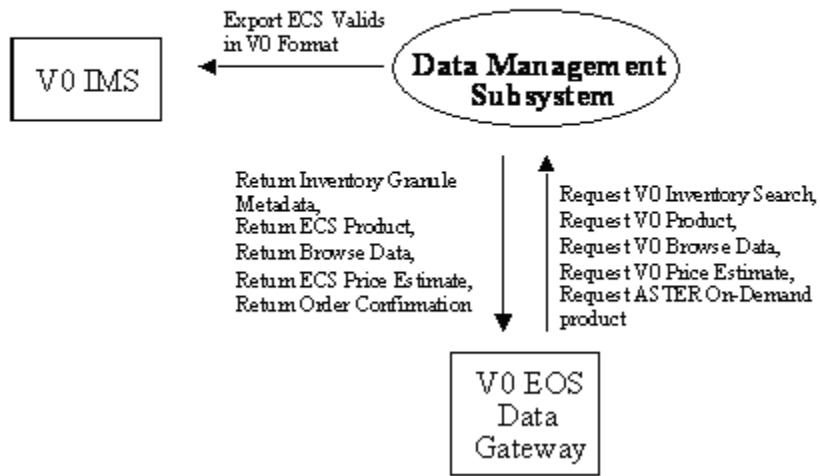
<http://delelln.gsfc.nasa.gov/~imswww/pub/manuals/imsdesign.html>

### **Data Management Subsystem Context**

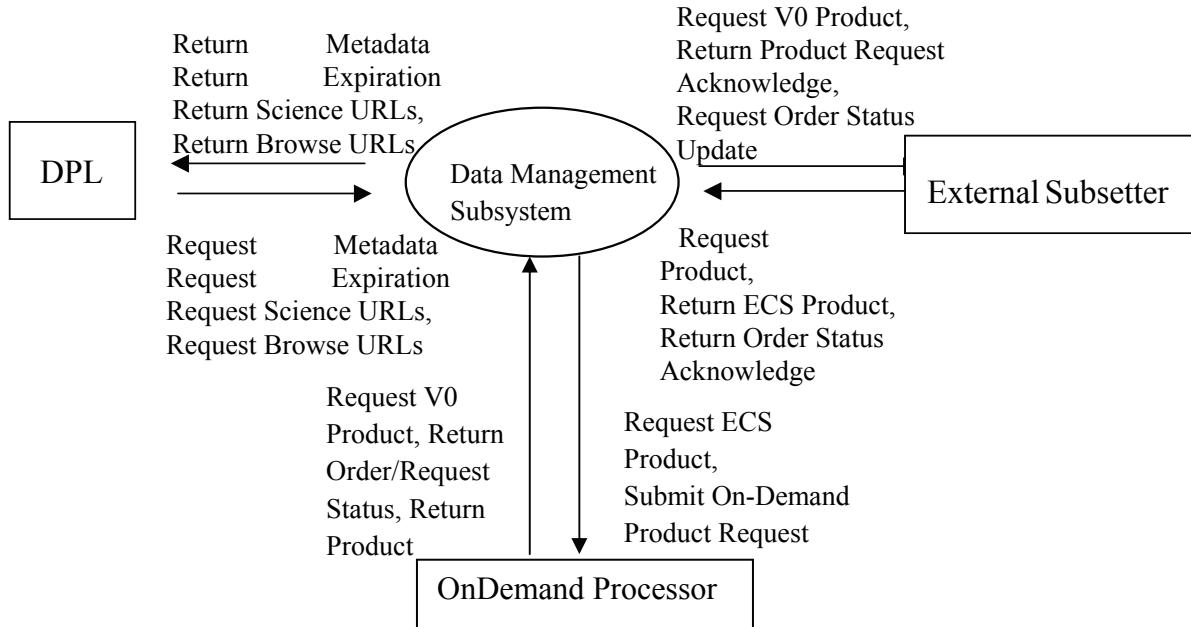
Figure 4.4-1 is the Data Management Subsystem context diagrams. The diagrams show the events sent to the Data Management Subsystem and the events the Data Management Subsystem sends to other CSMS subsystems. Table 4.4-1 provides descriptions of the interface events shown in the Data Management Subsystem context diagrams.



**Figure 4.4-1. Data Management Subsystem Context Diagram**



**Figure 4.4-1. Data Management Subsystem Context Diagram (cont.)**



**Figure 4.4-1. Data Management Subsystem Context Diagram (cont.)**

**Table 4.4-1. Data Management Subsystem Interface Events (1 of 3)**

Event	Interface Event Description
Insert Product Distribution Request	The Data Management Subsystem (DMS) inserts product distribution requests in the Order Manager Data Base Management System within the <b>Order Manager Subsystem (OMS)</b> .
Request Management Services	<p>The <b>MSS</b> provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include:</p> <ul style="list-style-type: none"> <li>• <b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).</li> </ul> <p>The MSS also interfaces with other subsystems to perform the following:</p> <ul style="list-style-type: none"> <li>• <b>Order/Request Tracking</b> - The DMS interfaces with the <b>MSS</b> Order/Request tracking service to create and track a user product order.</li> <li>• <b>User Profile Request</b> - The <b>MSS</b> provides requesting subsystems with User Profile parameters such as e-mail address and shipping address to support their processing activities.</li> </ul>
Return Order Status	The <b>MSS</b> provides order ids and order status information for products requested by users.
Return User Profiles	The <b>MSS</b> returns user profile information requested by users or ECS processes.
Search Inventory	The DMS submits inventory search requests to the Science Data Server within the <b>DSS</b> .
Request Product	The DMS submits the product requests to the <b>DSS</b> to acquire data granules.
Request Browse Data	The DMS submits the browse requests to the <b>DSS</b> to obtain browse data to determine the type of product to order.
Request Inspect	The DMS sends a request for an inspection of granule metadata to the <b>DSS</b> in support of a price estimate request.
Send ESDT Installation Information	The <b>DSS</b> inserts new collection level information into the DMS Data Dictionary database as new Earth Science Data Types (ESDTs) are added.
Replace ESDT Installation Information	The <b>DSS</b> sends updated ESDT information to the DMS Data Dictionary whenever an ESDT is updated. This data consists of updated Inventory and Collection level metadata.
Return Product	The DMS receives products from the <b>DSS</b> based upon a product request.
Return Browse Data	The DMS receives browse data associated with a particular granule from the <b>DSS</b> .
Return Inventory Granule Metadata	The DMS receives the granule metadata identifying the scene within the granule based on an inventory search request from the <b>DSS</b> .
Return Granule URs	The DMS receives Earth Science Data Type (ESDT) Universal References (URs) for the requested granules from the <b>DSS</b> .
Retrieve Metadata Type Info	The <b>CSS</b> retrieves type information for qualifying metadata specified in a SIPS search request from the DMS.

**Table 4.4-1. Data Management Subsystem Interface Events (2 of 3)**

Event	Interface Event Description
Request Communications Support	<p>The <b>CSS</b> provides a library of services available to each CSMS subsystems. The subsystem services required to perform specific assignments are requested from the CSS. These services include:</p> <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• Name/Address Services</li> <li>• Password Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> <li>• Request Distribution Media Options from the Configuration Registry</li> <li>• Request External Subsetting Options from the Configuration Registry</li> <li>• Request ASTER On-Demand Options from the Configuration Registry</li> </ul>
Return Inventory Granule Metadata	<p>The DMS returns the inventory granule metadata identifying the scene within the granule based on an inventory search request to the user via the <b>V0 EOS Data Gateway</b>.</p>
Request V0 Inventory Search	<p>The DMS receives inventory search requests from the <b>V0 EOS Data Gateway (EDG)</b> on behalf of an external ECS user.</p>
Request V0 Product	<p>The DMS receives product requests from the <b>V0 EOS Data Gateway</b> and <b>External Processor*</b> on behalf of an external ECS user.</p>
Request V0 Browse Data	<p>The DMS receives browse data requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.</p>
Request V0 Price Estimate	<p>The user sends a price estimate request for an ECS product to the DMS via the <b>V0 EOS Data Gateway</b>.</p>
Request Aster On-Demand Product	<p>The DMS receives product requests for ASTER On-Demand products from the <b>V0 EOS Data Gateway</b>.</p>

*\*Note: For the purpose of this document, “External Processor” refers to either an External Subsetter (HAS) or an On-Demand Processor (S4PM), both of which are treated virtually the same by the DMS.*

**Table 4.4-1. Data Management Subsystem Interface Events (3 of 3)**

Event	Interface Event Description
Return ECS Product	The DMS returns ECS data to the user via the <b>V0 EOS Data Gateway</b> based upon a product request.
Return Browse Data	The DMS returns Browse data associated with a particular granule to the user via the <b>V0 EOS Data Gateway</b> .
Return ECS Price Estimate	The DMS returns a price estimate for a price estimate request to the user via the <b>V0 EOS Data Gateway</b> .
Return Order Confirmation	The <b>V0 EOS Data Gateway</b> receives order confirmation of an ASTER On-Demand request from the DMS.
Export ECS Valids in V0 Format	The DMS sends the valids with the ECS core and PSA Attributes to the <b>V0 IMS</b> .
Return Product Request Acknowledge	The DMS receives confirmation of a subsetted request from an <b>External Subsetter</b> .
Request Order Status Update	The DMS receives order status update requests provided by <b>External Subsetters</b> and submits the update to MSS
Request Subsetted Product	The DMS receives product requests for subsetted products and sends requests for the products to be provided by an <b>External Subsetter</b> .
Return Order Status Acknowledge	The DMS returns confirmation of an order status update request to the <b>External Subsetter</b> .
Submit On-Demand Product Request	The <b>On-Demand Processor</b> (e.g. S4PM) receives product requests from the DMS for ASTER On-Demand products.
Return Order/Request Status	The <b>On-Demand Processor</b> (e.g. S4PM) returns the product request status to the DMS.
Request Metadata URL	The DMS retrieves Metadata URLs for granules from the <b>Data Pool Sybase ASE</b> .
Request Expiration Date	The DMS retrieves Expiration Dates for granules from the <b>Data Pool Sybase ASE</b> .
Request Science URLs	The DMS retrieves Science URL(s) for granules from the <b>Data Pool Sybase ASE</b> .
Request Browse URLs	The DMS retrieves Browse URL(s) for granules from the <b>Data Pool Sybase ASE</b> .
Return Metadata URLs	The <b>Data Pool Sybase ASE</b> returns Metadata URLs to the DMS.
Return Expiration date	The <b>Data Pool Sybase ASE</b> returns Expiration Dates to the DMS.
Return Science URLs	The <b>Data Pool Sybase ASE</b> returns zero or more Science URL(s) to the DMS.
Return Browse URLs	The <b>Data Pool Sybase ASE</b> returns zero or more Browse URL(s) to the DMS.

## **Data Management Subsystem Structure**

The DMS is two CSCIs and one Hardware Configuration Item (HWCI):

- The Data Dictionary (DDICT) is a software configuration item. DDICT manages the definitions of data collections including the metadata, data domains (valid values), and data location. The Data Dictionary information is stored persistently in a Relational Database Management System (DBMS).
- The Version 0 Gateway (V0 GTWAY) is a software configuration item. The V0 GTWAY CSCI provides access to data and services between the SDSRV CSCI and the V0 IMS. V0 GTWAY services include inventory searches; requests for browse data, product requests, and price estimate requests.
- The DMS hardware comprises one hardware configuration item called the Data Management Hardware (DMGHW) CI. DMGHW provides the servers and workstations needed for all data management functions. The DMGHW provides processing and storage for the DDICT and V0 GTWAY CSCIs.

## **Use of COTS in the Data Management Subsystem**

**Note: The following RogueWave Libraries are currently delivered with custom code as static libraries. A separate installation of dynamic libraries is no longer required.**

- RogueWave's Tools.h++

The Tools.h++ class libraries are used by the DMS to provide basic functions and objects such as strings and collections.

- RogueWave's DBTools.h++

The DBTools.h++ C++ class libraries are used to interact with the Sybase database Structured Query Language (SQL) server. The use of DBTools buffers the DMS processes from the relational database used.

Other COTS products are also utilized to support the Data Management Subsystem.

- The ICS Builder Xcessory GUI

The Builder Xcessory GUI builder tool modifies the displays of the Data Dictionary Maintenance Tool (Mtool). The builder tool also generates the C++ code to produce the Mtool displays at run time. There is no operational component of Builder Xcessory needed at run-time.

- Sybase Adaptive Server Enterprise (ASE)

Sybase's ASE provides access for the Data Dictionary to insert, update, and delete Data Dictionary database information. The Sybase ASE must be running during operations for the Data Dictionary Server to execute, search, and update requests on the Data Dictionary database.

- CCS Middleware Client

CCS Middleware Client provides DMS with communications between other subsystems. CCS Middleware can reside on one or both sides of the interface. An instance must be installed on the platform where DMS resides. Although the CCS Middleware Client is part of CSS.

### Error Handling and processing

`EcUtStatus` is a class used throughout the ECS custom code for general error reporting. It is almost always used as a return value for functions and allows detailed error codes to be passed back up function stacks.

The DDICT and V0 GTWAY CSCIs use two main mechanisms for error handling.

#### 1. Return `EcUtStatus` values

Functions can return an `EcUtStatus` object, indicating success, failure or other detailed status, which corresponds to `EcUtStatus::OK`, `EcUtStatus::FAILED` and `EcUtStatus::DETAILED`.

For `V0ToEcsGateway`, the `EcUtStatus` values are reflected on the EDG to indicate the search or order status.

#### 2. Exceptions

Some functions (e.g., a main function) cannot return `EcUtStatus` values to indicate success or failure. These functions can catch some exceptions after a try block.

All error messages are sent to the ALOG file or debug log file.

For writing messages to the Applications Log (ALOG), the following functions are used:

`EcLg.LogError` sends a message to the ALOG at severity level 1. For example, `EcLg.LogError("Error", 0, "Failed to get Order ID from MSS. TransactionID=%s.",`

```
myTransactionId.data());
```

`EcLg.LogInformation` sends a message to the ALOG at severity level 3. For example, `EcLg.LogInformation("DmGwManagedServer", 0, "Server Shutdown");`

For writing messages to the debug log, the following macros are used:

`PF_STATUS` writes a message at a “log level” of 1 to the debug log. For example, `PF_STATUS cerr << "Unable to create inputNonAggregate" << endl;`

`PF_VERBOSE` writes a message at a “log level” of 2 to the debug log. For example, `PF_VERBOSE cerr << "~~~ Connect to Science Data Server FAILED."`

PF\_DEBUG writes a message at a “log level” of 3 to the debug log. For example, PF\_DEBUG cerr << “~~~ Failed to get Order ID from MSS.” << endl;

#### **4.4.1 Data Dictionary Software Description**

##### **4.4.1.1 Data Dictionary Functional Overview**

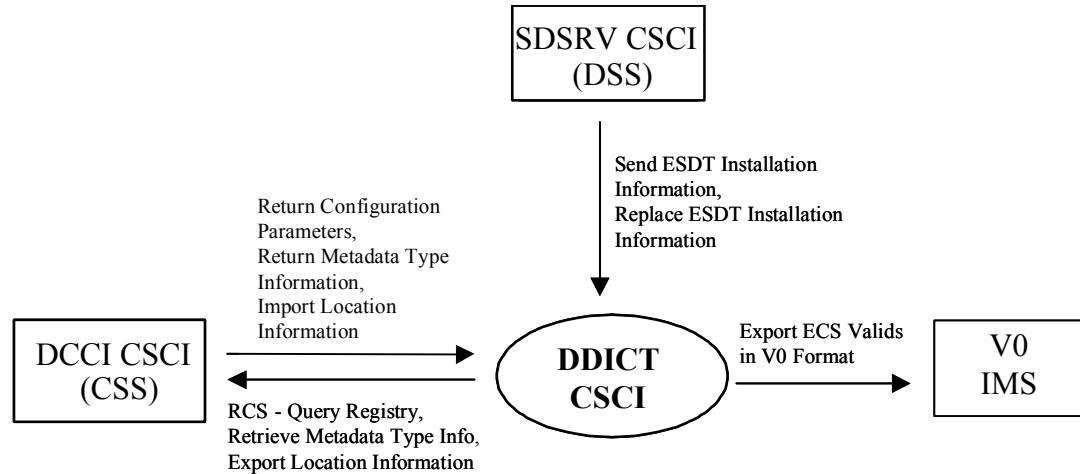
The Data Dictionary (DDICT) CSCI provides access to the Data Dictionary database containing information about science data collections, data attributes, data operations, and the domain(s) of the attributes. The DDICT CSCI describes the data objects accessible through Data Servers and the Gateways. The DDICT CSCI provides information support for users to retrieve definitions of the available items and provides infrastructure support to the other CSCIs within the DMS.

Clients (CSMS CSCIs, CSCs, or processes) of the Data Dictionary obtain data collection information by sending queries to the Data Dictionary. Mappings between the V0 IMS attributes and keywords are also maintained within the Data Dictionary. These mappings are used to translate requests between the V0 IMS and the ECS.

The location of a data collection within a data server at a particular site is also stored within the data dictionary. This information allows users to perform queries through user software such as the V0 EOS Data Gateway from any geographical location to forward inventory search, browse, and acquire requests to the appropriate Data Server or Gateway located at the site where the data is physically stored.

##### **4.4.1.2 Data Dictionary Context**

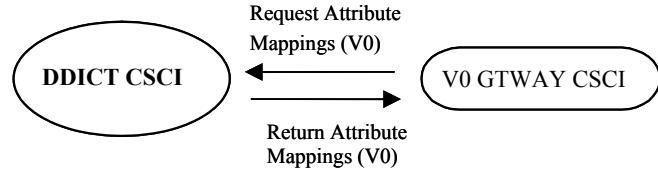
Figure 4.4-2 is the DDICT CSCI context diagrams. The diagrams show the events sent to other CSCIs or CSCs and the events the DDICT CSCI receives from other CSCIs and CSCs. Table 4.4-2 provides descriptions of the interface events shown in the DDICT CSCI context diagrams.



**Note:**

**ESDT** = Earth Science Data Type,  
**IMS** = Information Management System,  
**Info** = Information,  
**RCS** = Request Communications Support

**Figure 4.4-2. Data Dictionary CSCI Context Diagram**



**Note:**  
**V0** = Version 0,  
**DAAC** = Distributed Active Archive Center

**Figure 4.4-2. Data Dictionary CSCI Context Diagram (cont.)**

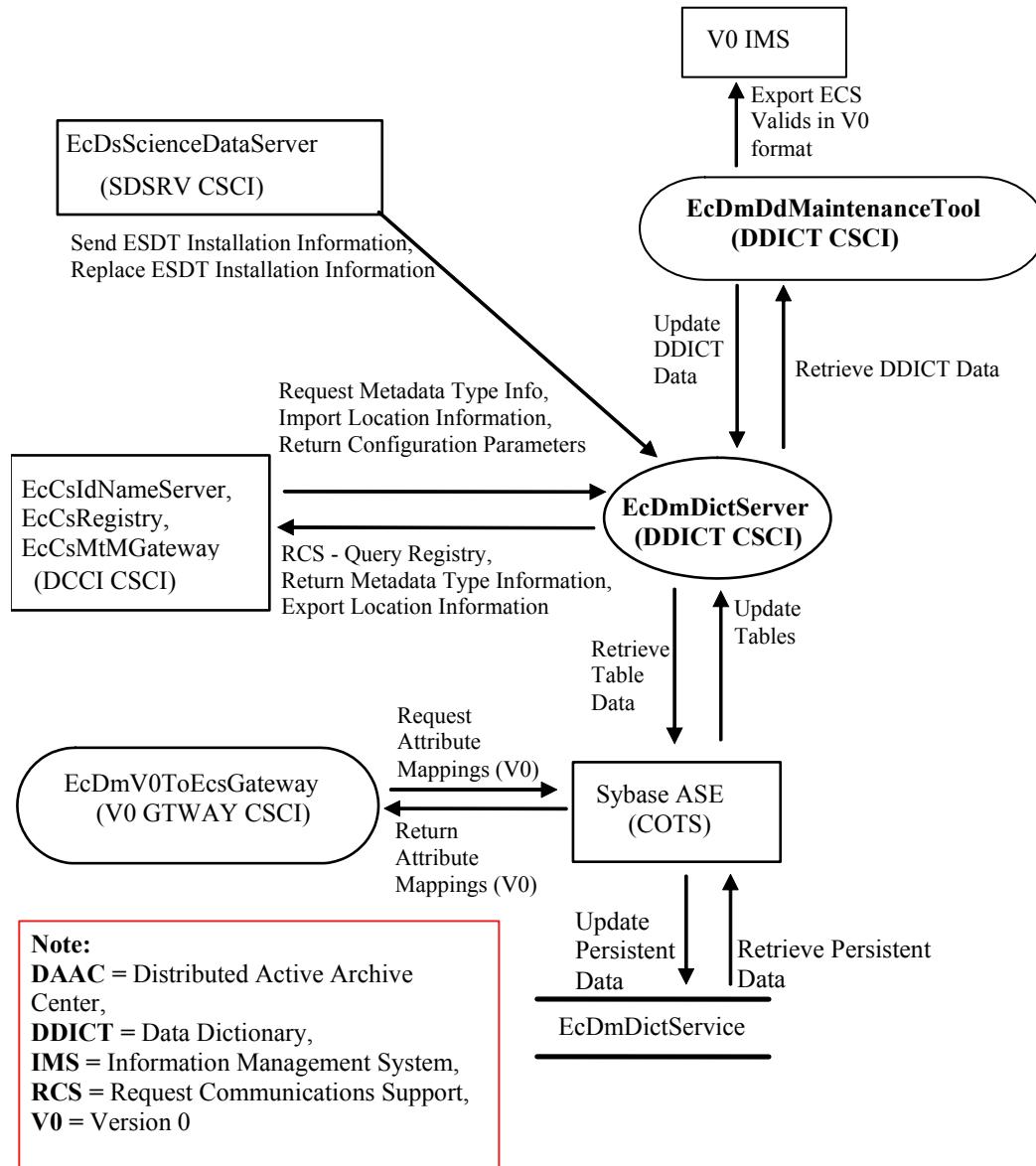
**Table 4.4-2. Data Dictionary CSCI Interface Events**

Event	Interface Event Description
Request Management Services	<b>System startup and shutdown</b> – Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).
Send ESDT Installation Information	The <b>SDSRV CSCI</b> inserts new collection level information into the DMS Data Dictionary database via the DDICT CSCI, as new Earth Science Data Types (ESDTs) are added.
Replace ESDT Installation Information	The <b>SDSRV CSCI</b> sends updated ESDT information to the DDICT CSCI whenever an ESDT is updated. This data consists of Inventory and Collection level metadata. The updated information replaces the ESDT information in the DDICT CSCI.
Request Communications Support	The <b>DCCI CSCI</b> provides a library of services available to each CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• Name/Address Services</li> <li>• Password Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry – Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> </ul>
Retrieve Metadata Type Info	The <b>DCCI CSCI</b> retrieves metadata type information from the DDICT CSCI pertaining to search or search & order requests from the SIPS.
Export Location Information	The DDICT CSCI stores physical and logical information in the <b>CCS NameServer</b> .
Return Configuration Parameters	The DDICT CSCI receives the configuration parameters and associated values from the Registry Server within the <b>DCCI CSCI</b> .
Return Metadata Type Info	The <b>DCCI CSCI</b> receives metadata type information from the <b>DDICT CSCI</b> .
Import Location Information	The DDICT CSCI requests server location information from the <b>CCS NameServer</b> .
Request Attribute Mappings (V0)	A user, via the EOS Data Gateway within the V0 IMS, requests data collection attribute and keyword mappings (via the <b>V0 GTWAY CSCI</b> ) from the DDICT CSCI to translate requests from the V0 IMS to the ECS and back again.
Return Attribute Mappings (V0)	The DDICT CSCI returns the data collection attribute and keyword mappings requested by the user via the <b>V0 GTWAY CSCI</b> .

#### 4.4.1.3 Data Dictionary Architecture

Figure 4.4-3 is the DDICT CSCI architecture diagram. The diagram shows the events sent to the DDICT CSCI processes and the events the DDICT CSCI processes send to other processes.

The DDICT CSCI is two processes, the Data Dictionary Server (EcDmDictServer) and the Data Dictionary Maintenance Tool (EcDmMaintenanceTool) and a COTS process (the Sybase ASE). The Data Dictionary Server, Maintenance Tool, and Sybase ASE processes reside inside a DAAC and run on the DMGHW. The Data Dictionary uses one data store per DAAC, the EcDmDictServer Database, as shown in Figure 4.4-3.



**Figure 4.4-3. Data Dictionary CSCI Architecture Diagram**

#### 4.4.1.4 Data Dictionary Process Descriptions

Table 4.4-3 provides descriptions of the processes shown in the Data Dictionary CSCI architecture diagram.

**Table 4.4-3. Data Dictionary CSCI Processes**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDmDictServer	Server	OMSHW	Developed	<p>The Data Dictionary Service is the primary server interface to collection and collection related information for the DMS and other subsystems. It allows DDICT client processes the capability to perform data searches, insertions, updates, or deletions to the collection information held in the DDICT database.</p> <p>The Data Dictionary offers two basic interfaces          DDICT Data Search: The Data Dictionary Server allows a user to specify search requests on the Data Dictionary database using a GIParameter List.</p> <p>DDICT Data Insert and Delete: Provides a client process with the capability to insert and delete data within the Data Dictionary database.</p> <p>The Data Dictionary Service supports:          Single requests at a time          Synchronous request processing          Asynchronous request processing</p>
EcDmDdMaintenanceTool	GUI	OMSHW	Developed	<p>Provides a graphical user interface (GUI) to insert, update, or delete schema information held in the DDICT database, allowing DAAC operations staff to maintain the data stored in the Data Dictionary database. The Data Dictionary Maintenance Tool also provides the following capabilities:</p> <p>Import and Export of Valid: The tool allows DAAC operations staff to import and export data collection attribute valids to and from ECS, and V0 IMS for catalog interoperability.</p> <p>Data Collection Attribute and Keyword Mapping: Allows DAAC operations staff to map data collection attributes and keyword valids from ECS to V0 IMS. The V0 GTWAY CSCI process (EcDmV0ToEcsGateway) that translates requests from ECS to V0 IMS uses this information.</p>
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE acts as a SQL server for the Data Dictionary, and is only run at the DAACs by DAAC operations staff. Refer to Sybase documentation for details.
EcDmValidsExporter	CLI	OMSHW	Developed	Export of Valid: This tool provides a command line mechanism for DAAC operators to export data collection attribute valids from ECS for catalog interoperability with V0 IMS.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

#### 4.4.1.5 Data Dictionary Process Interface Descriptions

Table 4.4-4 provides descriptions of the interface events shown in the Data Dictionary CSCI architecture diagram.

**Table 4.4-4. Data Dictionary CSCI Process Interface Events (1 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services	At system startup or shutdown and for restarts	<i>Process:</i> EcDmDictServer	DAAC unique startup scripts	<b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the ECS Project document (611) and the current ECS Project Training Material document (625), identified in Section 2.2.1 of this document.
Export ECS Validis in V0 Format	Based on V0 validis processing cycle ~ once per week	V0 IMS Host	DAAC Operations Staff <i>Process:</i> EcDmValidisExporter EcDmDdMaintenanceTool <i>Classes:</i> DmLmDbiUtils, gov.nasa.emd.dms.exportv alids.Exporter, DmDdMtMainWindow	The DAAC Operations Staff pushes the validis file, using the FTP service, manually to its destination location agreed upon earlier on a <b>V0 IMS Host</b> .
Retrieve DDICT data	One per Maintenance Tool search	<i>Process:</i> EcDmDdMaintenanceTool <i>Classes:</i> DmDdMtDBExtract, DmDdMtMainWindow	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCIRequest	The EcDmDictServer returns data collection information including collection lists, and collection attributes and keyword validis to the <b>EcDmDdMaintenanceTool</b> .
Update DDICT data	One per table information update	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCISchemaRequest	Operations Staff <i>Process:</i> EcDmDdMaintenanceTool <i>Classes:</i> DmLmDbiUtils, DmDdMtMainWindow	The Operations Staff, using the <b>EcDmDdMaintenanceTool</b> , updates table information within the DDICT database (via the EcDmDictServer) including mapping collection attributes to keywords and mapping collections to information managers and adding, modifying, and deleting Core Attributes.

**Table 4.4-4. Data Dictionary CSCI Process Interface Events (2 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Update Tables	One per database update	<i>Process:</i> Sybase ASE (COTS SW) RWDBTools.h++ classes	<i>Process:</i> EcDmDictServer <i>Library:</i> DmDdReqProc <i>Classes:</i> DmDdMapper, DmDdProcMsg	The EcDmDictServer receives updated data from the Data Dictionary database once the <b>Sybase ASE</b> inserts and deletes collections and collection metadata, attributes and keywords, and attribute and keyword mappings.
Retrieve Persistent Data	One per set number of queries	<i>Process:</i> Sybase ASE (COTS)	<i>Data Base Table:</i> EcDmDictService	The <b>Sybase ASE</b> retrieves data persistently stored on disk(s) based on search queries from the EcDmDictServer.
Update Persistent Data	One per set number of queries	<i>Data Base Table:</i> EcDmDictService	<i>Process:</i> Sybase ASE (COTS)	The <b>Sybase ASE</b> updates data persistently stored on disk(s) based on queries from the EcDmDictServer.
Request Attribute Mappings (V0)	One per request from V0 Gateway	<i>Process:</i> EcDmDictServer <i>COTS SW</i> <i>Library:</i> RWDBTools.h++ <i>Library:</i> DmLmDbi <i>Class:</i> DmLmIntQuery	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Persistent <i>Class:</i> DmGwAttributeMap	The <b>EcDmV0ToEcsGateway</b> requests data collection attribute and keyword mappings from the <b>EcDmDictServer</b> Data Dictionary database, via the Sybase ASE, to translate requests from the V0 IMS.
Return Attribute Mappings (V0)	One per request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Persistent <i>Class:</i> DmGwAttributeMap	<i>Process:</i> EcDmDictServer <i>COTS SW</i> <i>Library:</i> RWDBTools.h++ <i>Library:</i> DmLmDbi <i>Class:</i> DmLmIntQuery	The <b>EcDmDictServer</b> Data Dictionary database, via the Sybase ASE, returns the data collection attribute and keyword mappings requested by the user to the <b>EcDmV0ToEcsGateway</b> .
Retrieve Table Data	One per retrieve from the database	<i>Process:</i> Sybase ASE (COTS SW) RWDBTools.h++ classes	<i>Process:</i> EcDmDictServer <i>Library:</i> DmDdServerJit <i>Class:</i> DmDdSearchRequest	The EcDmDictServer requests data (such as collections and collection metadata, attributes and keywords, and attribute and keyword mappings) from the Data Dictionary database within the <b>Sybase ASE</b> .

**Table 4.4-4. Data Dictionary CSCI Process Interface Events (3 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	Request service(s) as required	<p><b>Process:</b> EcCsIdNameServer</p> <p><b>Libraries:</b> EcPf, Middleware, FoNs, Flop, oodce</p> <p><b>Classes:</b> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy</p> <p><b>Library (Common):</b> EcUr</p> <p><b>Class:</b> EcUrServerUR</p> <p><b>Library:</b> event</p> <p><b>Class:</b> EcLgErrorMsg</p> <p><b>Process:</b> EcCsRegistry</p> <p><b>Library:</b> EcCsRegistry</p> <p><b>Class:</b> EcRgRegistryServer_C</p>	<p><b>Process:</b> EcDmDictServer</p> <p><b>Libraries:</b> EcDmDdClient, DmDdReqProc, DmDdServerJit</p> <p><b>Classes:</b> DmDdCISchemaRequest, DmDdCIRequest, DmDdMapper, DmDdProcMsg, DmDdSearchRequest</p>	<p>The <b>DCCI CSCI</b> provides a library of services available to each CSMS process. The process services required to perform specific assignments are requested from the DCCI CSCI CSCs. These services include:</p> <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• Name/Address Services</li> <li>• Password Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> </ul>

**Table 4.4-4. Data Dictionary CSCI Process Interface Events (4 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return Metadata Type Info	Per search or order request	<i>Process:</i> EcCsMtMGateway <i>Class:</i> EcCsMtMAttributeDict	<i>Processes:</i> EcCsMtMGateway, EcDmDictServer <i>Libraries:</i> DmAsGwCommon, Common, DmGwV0Util, EcDmDdClient, DmDdMsg <i>Class:</i> DmDdCIRequestServer	The EcCsMtMGateway receives metadata type information from the <b>EcDmDictServer</b> .
Export Location Information	Once at system startup and after each failure recovery	<i>Process:</i> EcCsidNameServer <i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce <i>Classes:</i> EcPfManagedServer, CCSMdwnNameServer, FoNsNameServerProxy, CCSMdwnRwNetProxy	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCIRequest	The EcDmDictServer stores physical and logical location information in the <b>EcCsidNameServer</b> .
Request Metadata Type Info	One or more client request	<i>Processes:</i> EcCsMtMGateway, EcDmDictServer <i>Libraries:</i> DmAsGwCommon, Common, DmGwV0Util, EcDmDdClient, DmDdMsg <i>Class:</i> DmDdCIRequestServer	<i>Process:</i> EcCsMtMGateway <i>Class:</i> EcCsMtMAttributeDict	The EcCsMtMGateway requests metadata type information from the <b>EcDmDictServer</b> based upon the qualifying metadata contained in the requests sent by the SIPS via CCS Middleware calls.

**Table 4.4-4. Data Dictionary CSCI Process Interface Events (5 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	<i>Process:</i> EcDmDictServer <i>Libraries:</i> EcDmDdClient, DmDdReqProc, DmDdServerJit <i>Classes:</i> DmDdCISchemaRequest, DmDdCIRequest, DmDdMapper, DmDdProcMsg, DmDdSearchRequest	<i>Process:</i> EcCsIdNameServer <i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce <i>Classes:</i> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDmDictServer requests server location information from the <b>EcCsIdNameServer</b> .
Return Configuration Parameters	One set per request	<i>Process:</i> EcDmDictServer <i>Library:</i> DmDdServerLib <i>Class:</i> DmDdManagedServer	<i>Process:</i> EcCsRegistry <i>Library:</i> EcCsRegistry <i>Class:</i> EcRgRegistryServer_C	The <b>EcCsRegistry</b> returns the attribute-value pairs (configuration parameters) to the EcDmDictServer upon request.
Send ESDT Installation Information	One per new ESDT added to ECS	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCISchemaRequest	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsDe1 <i>Class:</i> DsDeDataDictController	The <b>EcDsScienceDataServer</b> inserts new collection information into the DMS Data Dictionary database when new Earth Science Data Types (ESDTs) are added.
Replace ESDT Installation Information	One per ESDT update	<i>Process:</i> EcDmDictServer <i>Library:</i> EcDmDdClient <i>Class:</i> DmDdCISchemaRequest	<i>Process:</i> EcDsScienceDataServer <i>Class:</i> DsDeDataDictController	The <b>EcDsScienceDataServer</b> sends updated ESDT information to the EcDmDictServer whenever an ESDT is updated. This data consists of updated Inventory and Collection level metadata.

#### **4.4.1.6 Data Dictionary CSCI Data Stores**

Table 4.4-5 provides descriptions of the data stores shown in the Data Dictionary CSCI architecture diagram.

**Table 4.4-5. Data Dictionary CSCI Data Stores**

Data Store	Type	Functionality
EcDmDictService	Database	<p>The Data Dictionary database, EcDmDictService is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium.</p> <p>Data stores in the Data Dictionary database include:</p> <ul style="list-style-type: none"><li>• Collection Types: A list of all the data types within the ECS.</li><li>• Collection Metadata: Various types of collection metadata including instrument, platform, sensor, topic, keyword, temporal and spatial data.</li><li>• Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the ECS.</li><li>• Collection Attribute and Keyword Mappings: Associations between the V0 IMS attributes and valid keywords and the ECS attributes and keywords are maintained.</li><li>• Collection to Information Manager Mappings: Associations between the information manager and the ECS collections stored within them are maintained.</li></ul>

#### **4.4.2 V0 Gateway Software Description**

##### **4.4.2.1 V0 Gateway Functional Overview**

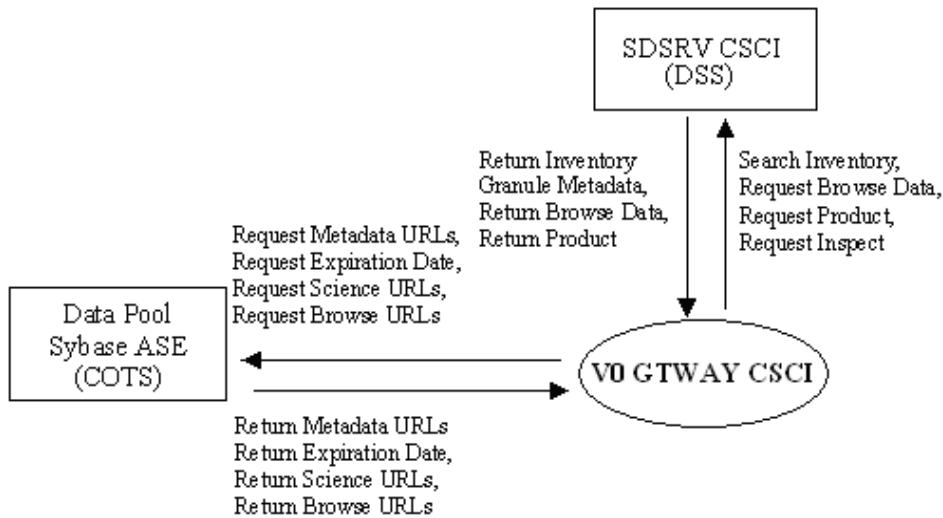
The Version 0 Gateway (V0 GTWAY) CSCI provides interoperability with the V0 Information Management System for inventory searches, browse requests, product orders, and price estimate requests.

The V0 GTWAY CSCI is one process, the V0 to ECS Gateway server. Queries are passed between the V0 IMS and the V0 Gateway processes using the Object Description Language (ODL) format. The structure of the V0 ODL messages is documented in “Interface Control Definition for the EOS Data Gateway (EDG) Messages and Development Data Dictionary” (423-42-06).

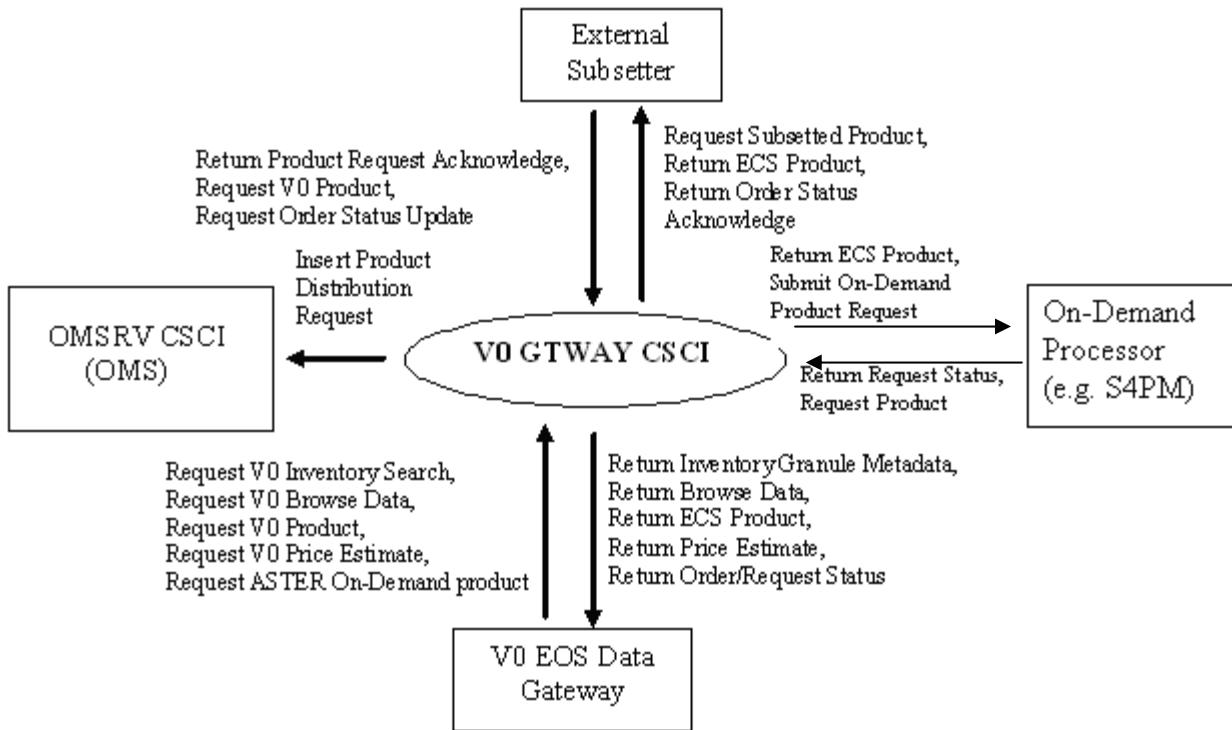
Since the V0 IMS uses different attributes to describe data collections within its data archive, the V0 GTWAY CSCI translates those attributes as defined in the system. To perform the translation, the V0 Gateway uses the data collection attribute and valid keyword mapping information contained within the Data Dictionary database to translate the V0 attributes into equivalent attributes.

#### 4.4.2.2 V0 Gateway Context

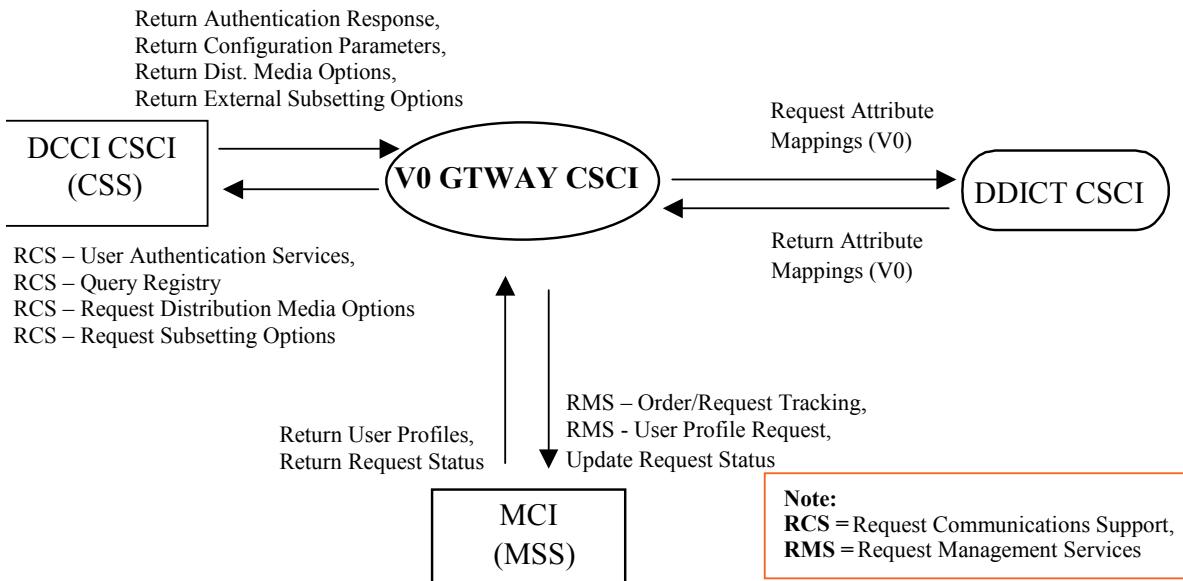
Figure 4.4-4 is the V0 GTWAY CSCI context diagrams. The diagrams show the events sent to the V0 GTWAY CSCI and the events the V0 GTWAY CSCI sends to other CSCIs. Table 4.4-6 provides descriptions of the interface events shown in the V0 GTWAY CSCI context diagrams.



**Figure 4.4-4. V0 GTWAY CSCI Context Diagram**



**Figure 4.4-4. V0 GTWAY CSCl Context Diagram (cont.)**



**Figure 4.4-4. V0 GTWAY CSCl Context Diagram (cont.)**

**Table 4.4-6. V0 GTWAY CSCI Interface Events (1 of 4)**

Event	Interface Event Description
Search Inventory	The V0 GTWAY CSCI submits inventory search requests to the <b>SDSRV CSCI</b> within the DSS on behalf of a user.
Request Browse Data	The V0 GTWAY CSCI receives browse requests from the V0 IMS via the EOS Data Gateway and submits the browse requests to the <b>SDSRV CSCI</b> within the DSS on behalf of a user.
Request Product	The V0 GTWAY CSCI submits product requests to the <b>SDSRV CSCI</b> within the DSS on behalf of a user.
Request Inspect	The V0 GTWAY CSCI sends a request for an inspection of granule metadata to the <b>SDSRV CSCI</b> in support of a price estimate request.
Return Status of User Billing Information	The <b>EDC DORRAN (Billing and Accounting System)</b> returns the status of the user billing information to the V0 GTWAY CSCI to allow or deny the user access to ECS data and services.
Return Metadata URLs	The <b>Data Pool Sybase ASE</b> returns Metadata URLs to the V0 GTWAY CSCI.
Return Expiration Date	The <b>Data Pool Sybase ASE</b> returns an Expiration Date to the V0 GTWAY CSCI.
Return Science URLs	The <b>Data Pool Sybase ASE</b> returns zero or more Science URL(s) to the V0 GTWAY CSCI.
Return Browse URLs	The <b>Data Pool Sybase ASE</b> returns zero or more Browse URL(s) to the V0 GTWAY CSCI.
Request Metadata URLs	The V0 GTWAY CSCI attempts to retrieve Metadata URLs from the <b>Data Pool Sybase ASE</b> .
Request Expiration Date	The V0 GTWAY CSCI attempts to retrieve Expiration Dates from the <b>Data Pool Sybase ASE</b> .
Request Science URLs	The V0 GTWAY CSCI attempts to retrieve Science URLs from the <b>Data Pool Sybase ASE</b> .
Request Browse URLs	The V0 GTWAY CSCI attempts to retrieve Browse URLs from the <b>Data Pool Sybase ASE</b> .
Return Inventory Granule Metadata	The V0 GTWAY CSCI receives the inventory granule metadata identifying the scene within the granule based on an inventory search request from the <b>SDSRV CSCI</b> .
Return Granule URs	The V0 GTWAY CSCI receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the <b>SDSRV CSCI</b> .
Return Browse Data	The V0 GTWAY CSCI receives browse data associated with a particular granule from the <b>SDSRV CSCI</b> .
Return Product	The V0 GTWAY CSCI receives granules from the <b>SDSRV CSCI</b> based upon a product request.
Request Subsetted Product	The V0 GTWAY CSCI receives product requests for subsetted products and requests the product be provided from an <b>External Subsetter</b> .
Return ECS Product	The V0 GTWAY CSCI returns granules (including Landsat 7 data archived within the ECS) to the <b>V0 EOS Data Gateway (EDG)</b> .
Return Order Status Acknowledge	The V0 GTWAY CSCI returns status of the order status update to an <b>External Subsetter</b> .
Submit On-Demand Product Request	The <b>On-Demand Processor</b> receives product requests from the V0 GTWAY CSCI for ASTER On-Demand products.

**Table 4.4-6. V0 GTWAY CSCI Interface Events (2 of 4)**

Event	Interface Event Description
Return Request Status	The <b>On-Demand Processor</b> returns the status of the product request to the V0 GTWAY CSCI. The <b>MCI</b> returns the status of the update request to the V0 GTWAY CSCI.
Return Inventory Granule Metadata	The V0 GTWAY CSCI returns the inventory granule metadata identifying the scene within the granule to the <b>V0 EOS Data Gateway (EDG)</b> .
Return Browse Data	The V0 GTWAY CSCI returns the browse data associated with a particular granule to the <b>V0 EOS Data Gateway</b> .
Return ECS Product	The V0 GTWAY CSCI returns granules to the <b>V0 EOS Data Gateway</b> .
Return Price Estimate	The V0 GTWAY CSCI returns a price estimate for a price estimate request to the user via the <b>V0 EOS Data Gateway</b> .
Return Order/Request Status	The <b>V0 EOS Data Gateway</b> receives order and request status for ASTER On-Demand products from the V0 GTWAY CSCI.
Request V0 Inventory Search	The V0 GTWAY CSCI receives inventory search requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.
Request V0 Browse Data	The V0 GTWAY CSCI receives browse data requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.
Request V0 Product	The V0 GTWAY CSCI receives product requests from the <b>V0 EOS Data Gateway</b> or an <b>External Processor</b> on behalf of an external ECS user.
Request V0 Price Estimate	The V0 GTWAY CSCI receives price estimate requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.
Request ASTER On-Demand product	The V0 GTWAY CSCI receives product requests for ASTER On-Demand products from the <b>V0 EOS Data Gateway</b> .
Insert Product Distribution Request	The V0 Gateway (V0 GTWAY) inserts product distribution requests into the Order Manager Data Base Management System (DBMS) within the <b>Order Manager Subsystem (OMS)</b> .
Return Product Request Acknowledge	The V0 GTWAY CSCI receives confirmation of a subsetted request from an <b>External Subsetter</b> .
Request Order Status Update	The V0 GTWAY CSCI receives order status update requests provided by an <b>External Subsetter</b> .
Request Attribute Mappings (V0)	The V0 GTWAY CSCI sends requests for data collection attribute and keyword mappings, on behalf of a user, to the <b>DDICT CSCI</b> Data Dictionary database via the Sybase ASE to translate requests from the V0 IMS to the ECS protocol and back again.
Return Attribute Mappings (V0)	The <b>DDICT CSCI</b> Data Dictionary database, via the Sybase ASE, returns the data collection attribute and keyword mappings to the V0 GTWY CSCI on behalf of a user.

**Table 4.4-6. V0 GTWAY CSCI Interface Events (3 of 4)**

Event	Interface Event Description
Request Management Services	<p>The <b>MCI</b> provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes:</p> <ul style="list-style-type: none"> <li>• <b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).</li> </ul> <p>The MCI also interfaces with other CSCIs to perform the following:</p> <ul style="list-style-type: none"> <li>• <b>Order/Request Tracking</b> - The V0 GTWAY CSCI interfaces with the <b>MCI</b> Order/Request tracking service (EcMsAcOrderSrvr) to create a user product order.</li> <li>• <b>User Profile Request</b> - The <b>MCI</b> provides requesting CSCIs with User Profile parameters such as e-mail address and shipping address to support their processing activities.</li> </ul>
Update Request Status	The V0 GTWAY CSCI receives order status update requests provided by an <b>External Subsetter</b> and submits the update to the MSS.
Return User Profiles	The V0 GTWAY CSCI receives user profile information from the <b>MCI</b> to authenticate a user.
Request Communications Support	<p>The <b>DCCI CSCI</b> provides a library of services available to each CSMS CSCI. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include:</p> <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• File Transfer Services</li> <li>• Network &amp; Distributed File Services</li> <li>• Bulk Data Transfer Services</li> <li>• Name/Address Services</li> <li>• Password Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> <li>• Request Distribution Media Options from the Configuration Registry</li> <li>• Request External Subsetting Options from the Configuration Registry</li> <li>• Request ASTER On-Demand Options from the Configuration Registry</li> </ul>
Return Authentication Response	The V0 GTWAY CSCI receives the response to authenticate the user from the <b>DCCI CSCI</b> .

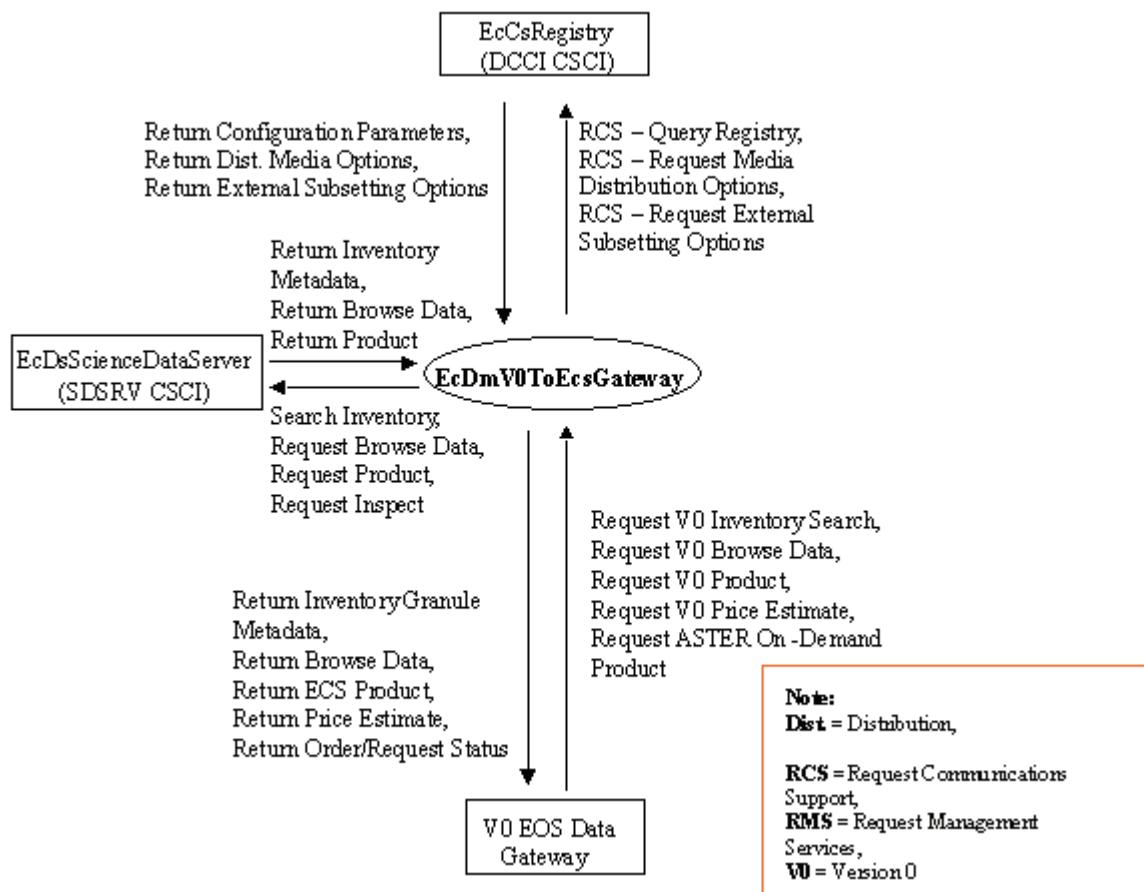
**Table 4.4-6. V0 GTWAY CSCI Interface Events (4 of 4)**

Event	Interface Event Description
Return Configuration Parameters	The V0 Gateway CSCI receives the configuration parameters and associated values from the Registry Server within the <b>DCCI CSCI</b> .
Return Dist. Media Options	The V0 GTWAY CSCI receives the requested distribution media options from the <b>DCCI CSCI</b> .
Return External Subsetting Options	The V0 GTWAY CSCI receives the requested external subsetting options from the <b>DCCI CSCI</b> .

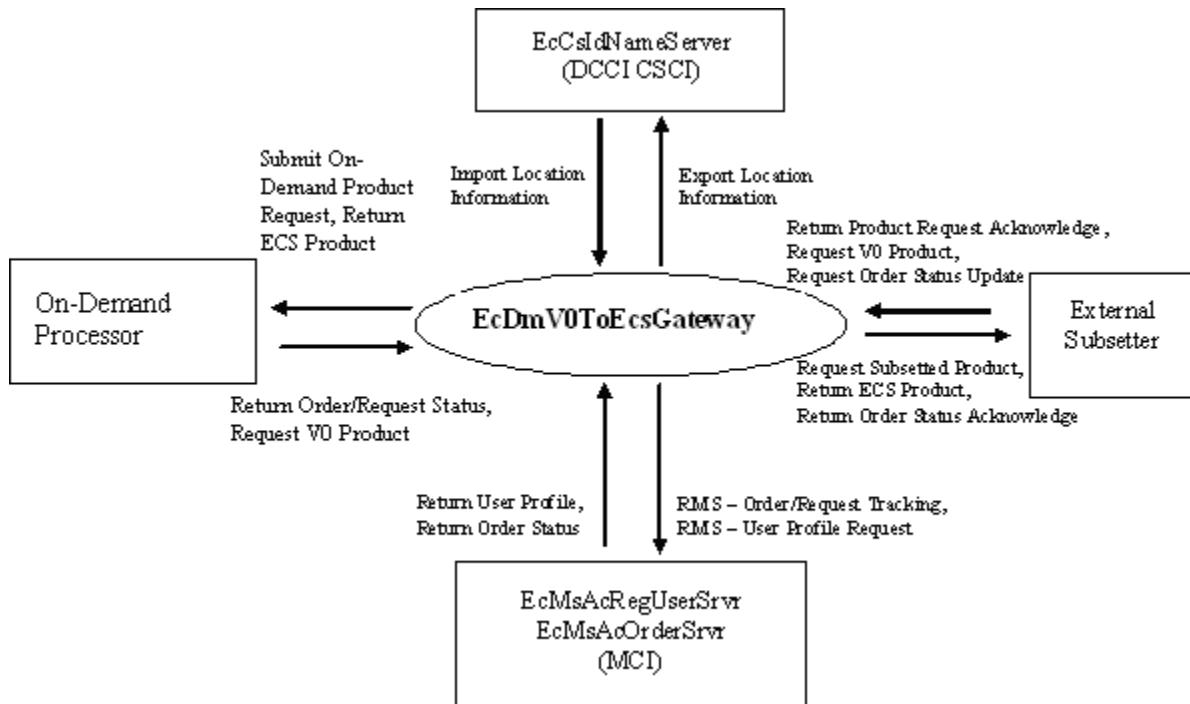
#### 4.4.2.3 V0 Gateway Architecture

Figure 4.4-5 is the V0 GTWAY CSCI architecture diagram. The diagram shows the events sent to the V0 GTWAY CSCI processes and the events the V0 GTWAY CSCI processes send to other processes.

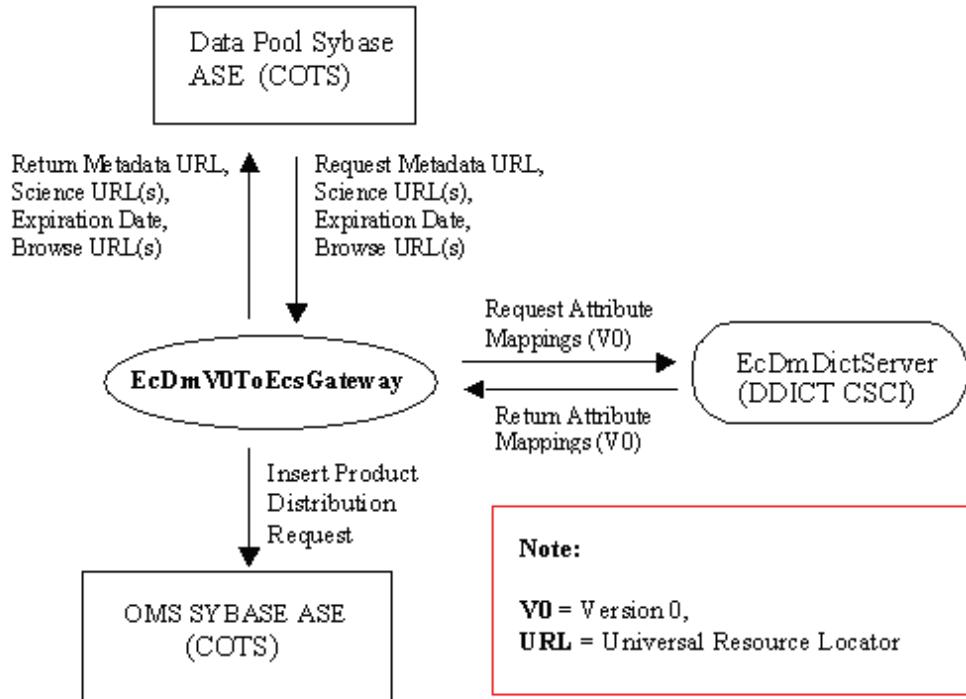
The V0 GTWAY CSCI is one process, the EcDmV0ToEcsGateway, as shown in the V0 GTWAY CSCI architecture diagram.



**Figure 4.4-5. V0 GTWAY CSCI Architecture Diagram**



**Figure 4.4-5. V0 GTWAY CSCI Architecture Diagram (cont.)**



**Figure 4.4-5. V0 GTWAY CSCI Architecture Diagram (cont.)**

#### 4.4.2.4 V0 Gateway Process Descriptions

Table 4.4-7 provides descriptions of the processes shown in the V0 GTWAY CSCI architecture diagrams.

**Table 4.4-7. V0 GTWAY CSCI Processes**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDmV0ToEcsGateway	Server	INTHW	Developed	<p>The V0 to ECS Gateway server allows users of the V0 IMS to query on data and services defined.</p> <p>Major Interfaces:</p> <ul style="list-style-type: none"><li>• Inventory Search: Allows a user to perform searches for data granules within the archive. URLs to metadata, browse and science granules stored within the Data Pool are also returned along with the expiration date of those URLs.</li><li>• Browse: Allows users to browse data granules previously found during a search</li><li>• Product request: Provides the capability for users to submit data acquire requests for obtaining data granules from the archive</li><li>• Order Status Update Request: Provides the capability for External Processors to provide status updates of orders being fulfilled by the Processor but submitted to ECS</li><li>• Price Estimate requests: Allows users to request a price estimate for a given set of granules with spatial and/or band sub-setting constraints</li><li>• Inspect Requests: Requests DSS to inspect granule metadata in support of a price estimate request.</li><li>• ASTER On-Demand: Allows a user to do a product request on ASTER On-Demand products</li></ul> <p>Server Supports:</p> <ul style="list-style-type: none"><li>• Synchronous request processing</li><li>• Asynchronous request processing</li><li>• Multiple concurrent requests</li></ul>

#### **4.4.2.5 V0 Gateway Process Interface Descriptions**

Table 4.4-8 provides descriptions of the interface events shown in the V0 GTWAY CSCI architecture diagrams.

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (1 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	Per process request	<p><i>Process:</i> EcCsIdNameServer</p> <p><i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce</p> <p><i>Classes:</i> EcPfManagedServer, EcPfClient, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy</p> <p><i>Library (Common):</i> EcUr</p> <p><i>Class:</i> EcUrServerUR</p> <p><i>Library:</i> event</p> <p><i>Class:</i> EcLgErrorMsg</p> <p><i>Process:</i> EcCsRegistry</p> <p><i>Library:</i> EcCsRegistry</p> <p><i>Class:</i> EcRgRegistryServer_C</p>	<p><i>Process:</i> EcDmV0ToEcsGateway</p> <p><i>Library:</i> Common</p> <p><i>Class:</i> DmGwManagedServer</p>	<p>The <b>DCCI CSCI</b> provides a library of services available to each CSMS process. The process services required to perform specific assignments are requested from the DCCI CSCI. These services include:</p> <ul style="list-style-type: none"> <li>• CCS Middleware Support</li> <li>• Database Connection Services</li> <li>• File Transfer Services</li> <li>• Network &amp; Distributed File Services</li> <li>• Bulk Data Transfer Services</li> <li>• Name/Address Services</li> <li>• Password Services</li> <li>• Server Request Framework (SRF)</li> <li>• Universal Reference (UR)</li> <li>• Error/Event Logging</li> <li>• Fault Handling Services</li> <li>• Mode Information</li> <li>• Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry</li> <li>• Request Distribution Media Options from the Configuration Registry</li> <li>• Request External Subsetting Options from the Configuration Registry</li> <li>• Request ASTER On-Demand Options from the Configuration Registry</li> </ul>

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (2 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request V0 Inventory Search	One per inventory search request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwInventoryRequest	<i>Process:</i> V0 EOS Data Gateway	The EcDmV0ToEcsGateway receives inventory search requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.
Request V0 Browse Data	One per browse request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwBrowseRequest	<i>Process:</i> V0 EOS Data Gateway	The EcDmV0ToEcsGateway receives browse data requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.
Request V0 Product	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> V0 EOS Data Gateway  <i>Process:</i> External Subsetter <i>Library:</i> Ik <i>Class:</i> IK_RxODL	The EcDmV0ToEcsGateway receives product requests from the <b>V0 EOS Data Gateway</b> or <b>External Processor</b> on behalf of an external ECS user.

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (3 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request V0 Price Estimate	One per product	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwPriceEstimateRequest	<i>Process:</i> V0 EOS Data Gateway	The EcDmV0ToEcsGateway receives price estimate requests from the <b>V0 EOS Data Gateway</b> on behalf of an external ECS user.
Submit On-Demand Product Request	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	<b><i>Process:</i></b> V0 EOS Data Gateway <b><i>Process:</i></b> On-Demand Processor	The EcDmV0ToEcsGateway receives product requests from the <b>V0 EOS Data Gateway</b> for ASTER On-Demand products. Aster On-Demand requests are processed virtually the same as external subsetter requests and sent to an external processor.
Return Inventory Granule Metadata	One per requested granule	<i>Process:</i> V0 EOS Data Gateway  <i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwInventoryRequest	The EcDmV0ToEcsGateway forwards the inventory granule metadata identifying the scene within the granule based on an inventory search request from the <b>V0 EOS Data Gateway</b> or the <b>EcDsScienceDataServer</b> .
Return Browse Data	One per request	<i>Process:</i> V0 EOS Data Gateway  <i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwV0BrowseRequest	The EcDmV0ToEcsGateway receives browse data associated with a particular granule from the <b>V0 EOS Data Gateway</b> or the <b>EcDsScienceDataServer</b> .

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (4 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return ECS Product	One per product request	<i>Process:</i> V0 EOS Data Gateway <i>Process:</i> External Subsetter <i>Library:</i> Ik <i>Class:</i> IK_RxODL	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwProductRequest	The EcDmV0ToEcsGateway receives products based upon V0 user product requests from the <b>V0 EOS Data Gateway</b> or an <b>External Processor</b> .
Return Price Estimate	One estimate per order	<i>Process:</i> V0 EOS Data Gateway	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwPriceEstimateRequest	The EcDmV0ToEcsGateway calculates the price estimate and sends the estimate to the user via the <b>V0 EOS Data Gateway</b> .
Return Order/ Request Status	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> V0 EOS Data Gateway	The EcDmV0ToEcsGateway returns an order status and request status acknowledgment back to the <b>V0 EOS Data Gateway</b> .
Search Inventory	One per inventory search request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwInventoryRequest	The EcDmV0ToEcsGateway submits inventory search requests to the <b>EcDsScienceDataServer</b> on behalf of a user.
Request Browse Data	One per browse request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwBrowseRequest	The EcDmV0ToEcsGateway submits requests for browse data to the <b>EcDsScienceDataServer</b> on behalf of a user.

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (5 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Request Product	One per product request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	The EcDmV0ToEcsGateway submits product requests to the <b>EcDsScienceDataServer</b> on behalf of a user.
Request Inspect	One per price estimate request	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwPriceEstimateRequest	The EcDmV0ToEcsGateway sends a request for an inspection of granule metadata to the <b>EcDsScienceDataServer</b> in support of a price estimate request.
Return Product	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> EcDsScienceDataServer <i>Library:</i> DsCI <i>Classes:</i> DsCIESDTReferenceCollector, DsCIRequest	The <b>EcDsScienceDataServer</b> returns products based upon a product request from a user.
Return Configuration Parameters	One set per request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Common <i>Class:</i> DmGwConfigItems	<i>Process:</i> EcCsRegistry <i>Library:</i> EcCsRegistry <i>Class:</i> EcRgRegistryServer_C	The <b>EcCsRegistry</b> returns the attribute-value pairs (configuration parameters) to the EcDmV0ToEcsGateway upon request.
Return Dist. Media Options	One set per request	<i>Process:</i> EcCsRegistry <i>Library:</i> EcCsRegistry <i>Class:</i> EcRgRegistryServer_C	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Common <i>Class:</i> DmGwConfigItems	The <b>EcCsRegistry</b> returns the requested distribution media options to the EcDmV0ToEcsGateway.

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (6 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return External Subsetting Options	One set per request	<i>Process:</i> EcCsRegistry <i>Library:</i> EcCsRegistry <i>Class:</i> EcRgRegistryServer_C	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Common <i>Class:</i> DmGwConfigItems	The <b>EcCsRegistry</b> returns the requested external processing options to the EcDmV0ToEcsGateway.
Export Location Information	Once at system startup and after each failure recovery	<i>Process:</i> EcCslNameServer <i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce <i>Classes:</i> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Common <i>Class:</i> DmGwManagedServer	The EcDmV0ToEcsGateway stores physical and logical server location information in the <b>EcCslNameServer</b> .
Return Product Request Acknowledge	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> External Processor <i>Library:</i> Ik <i>Class:</i> IK_RxODL	The EcDmV0ToEcsGateway receives a product result from an <b>External Processor</b> upon submission of a request for a subsetted product.
Request Order Status Update	One to three per subsetted product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwOrderStatusUpdate Request	<i>Process:</i> External Processor <i>Library:</i> Ik <i>Class:</i> IK_RxODL	The EcDmV0ToEcsGateway receives order status update requests from an <b>External Processor</b> . 1. Being Processed 2. Completed Processing or Terminated 3. Shipped
Request Subsetted Product	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> External Processor <i>Library:</i> Ik <i>Class:</i> IK_RxODL	The EcDmV0ToEcsGateway receives product requests from the <b>External Processor</b> for subsetted products on behalf of an external ECS user.

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (7 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Return Order Status Acknowledgment	One per Order Status Update request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> RequestProcessing <i>Class:</i> DmGwOrderStatusUpdate Request	<i>Process:</i> External Processor <i>Library:</i> Ik <i>Class:</i> IK_RXODL	The EcDmV0ToEcsGateway returns an order status acknowledgment upon receipt of an order status update request from an <b>External Processor</b> .
Return User Profile	One per profile request	<i>Process:</i> EcDmV0ToEcsGateway <i>Class:</i> DmGwRequestReceiver	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Class:</i> EcAcProfileMgr	The <b>EcMsAcRegUserSrvr</b> returns a user profile based upon a profile request from a user.
Return Order Status	One per order request	<i>Process:</i> EcDmV0ToEcsGateway <i>Class:</i> DmGwRequestReceiver	<i>Process:</i> EcMsAcOrderSrvr <i>Library:</i> MsAcClnt <i>Class:</i> EcAcOrderCMgr	The <b>EcMsAcOrderSrvr</b> returns an order id and status to the EcDmV0ToEcsGateway to send back to the user.
Return Order/Request Status	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Request Processing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> On-Demand Processor <i>Library:</i> PIOdMgrClient <i>Class:</i> PIOdMsgProxy	The EcDmV0ToEcsGateway receives order/request status for ASTER On-Demand product request from the On-Demand Processor.
Submit On-Demand Product Request	One per product request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Request Processing <i>Class:</i> DmGwAcquireRequest	<i>Process:</i> V0 EOS Data Gateway	The EcDmV0ToEcsGateway sends product requests to the On-Demand Processor for ASTER On-Demand products

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (8 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	<i>Process:</i> EcDmV0ToEcsGateway <i>Class:</i> DmGwRequestReceiver	<i>Process:</i> EcCsIdNameServer <i>Libraries:</i> EcPf, Middleware, FoNs, Folp, oodce <i>Classes:</i> EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The EcDmV0ToEcsGateway requests server location information from the <b>EcCsIdNameServer</b> .
Request Metadata URL, Science URLs, Expiration Date, Browse URLs	One per granule	Data Pool Sybase ASE (COTS) <i>Library:</i> RWDb <i>Class:</i> RWDBManager	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Common <i>Class:</i> DmGwDatapoolProxy	The EcDmV0ToEcsGateway retrieves metadata URL, science URLs, browse URLs and expiration dates from the <b>Sybase ASE</b> .
Request Attribute Mappings (V0)	One set per request	<i>Process:</i> EcDmDictServer, <i>COTS SW Library:</i> RWDBTools.h++ <i>Library:</i> DmLmDbi <i>Class:</i> DmLmIntQuery	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Persistent <i>Class:</i> DmGwAttributeMap	The EcDmV0ToEcsGateway sends requests to the <b>EcDmDictServer Data Dictionary database via the Sybase ASE</b> for data collection and keyword mappings to translate requests from the V0 IMS.
Return Attribute Mappings (V0)	One set per request	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Persistent <i>Class:</i> DmGwAttributeMap	<i>Process:</i> EcDmDictServer <i>COTS SW</i> <i>Library:</i> RWDBTools.h++ <i>Library:</i> DmLmDbi <i>Class:</i> DmLmIntQuery	The EcDmV0ToEcsGateway receives data collection and keyword mappings to translate requests from the V0 IMS to the <b>EcDmDictServer Data Dictionary database via the Sybase ASE</b> .

**Table 4.4-8. V0 GTWAY CSCI Process Interface Events (9 of 9)**

Event	Event Frequency	Interface	Initiated By	Event Description
Insert Product Distribution Request	One per service request	<i>Process:</i> Sybase Server (COTS)	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> OmClientlib <i>Classes:</i> OmSrDbInterface, OmSrV0InputIf	The EcDmV0ToEcsGateway inserts product distribution requests into the Order Manager DBMS.
Return Metadata URL, Science URL(s), Expiration Date, Browse URL(s)	One per granule	<i>Process:</i> EcDmV0ToEcsGateway <i>Library:</i> Common <i>Class:</i> DmGwDatapoolProxy	Data Pool Sybase ASE (COTS) <i>Library:</i> RWDb <i>Class:</i> RWDBManager	The <b>Sybase ASE</b> returns the Metadata URL, Science URL(s), Browse URL(s) and Expiration Dates to the EcDmV0ToEcsGateway.
Request Management Services (RMS)	At system startup or shutdown and for restarts	<i>Process:</i> EcDmV0ToEcsGateway	DAAC unique startup scripts	<b>System startup and shutdown</b> - Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).
RMS (Cont.)	One per product order	<i>Process:</i> EcMsAcOrderSrvr <i>Library:</i> MsAcClnt <i>Class:</i> EcAcOrderCMgr	<i>Process:</i> EcDmV0ToEcsGateway <i>Class:</i> DmGwRequestReceiver	<b>DMS Order/Request Tracking Update</b> – The V0 GTWAY interfaces with the Order/Request tracking service, <b>EcMsAcOrderSrvr</b> , to create a user product order.
RMS (Cont.)	One per profile request	<i>Process:</i> EcMsAcRegUserSrvr <i>Library:</i> MsAcClnt <i>Class:</i> EcAcProfileMgr	<i>Process:</i> EcDmV0ToEcsGateway <i>Class:</i> DmGwRequestReceiver	<b>User Profile Request</b> - The <b>EcMsAcRegUserSrvr</b> provides requesting processes with User Profile information such as e-mail address and shipping address to support their processing activities.

#### 4.4.2.6 V0 Gateway Data Stores

Table 4.4-9 provides descriptions of the data stores shown in the V0 GTWAY CSCI architecture diagram. The V0 to ECS and ECS to V0 Gateway processes access the Data Dictionary data store.

**Table 4.4-9. V0 GTWAY CSCI Data Store**

Data Store	Type	Functionality
EcDmDictService	Database	<p>The Data Dictionary database, EcDmDictService, is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium. The DDICT database is replicated wholly to each DAAC.</p> <p>The data stores in the Data Dictionary database used by the V0 GTWAY CSCI are:</p> <ul style="list-style-type: none"><li>• Collection Types: A list of all the data types</li><li>• Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the system are used by the V0 GTWAY CSCI to translate requests from the V0 IMS.</li></ul>

#### **4.4.3 Data Management Subsystem Hardware**

The primary components of the Data Management Subsystem include two hardware CIs, Data Management Hardware CI (DMGHW) and Interface Hardware CI (INTHW), co-owned by the Interoperability Subsystem, as described below. The general-purpose workstations are standalone hosts without fail-over capability. In the event of a host failure, any of the available workstations could be used to support end user DAAC maintenance.

##### **4.4.3.1 Data Management Hardware CI (DMGHW) Description**

The DMGHW CI includes Linux workstations. In the EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping. These workstations are used as end user workstations in maintenance of each of the respective DAAC sites.

##### **4.4.3.2 Interface Hardware CI (INTHW) Description, as used by the Data Management Subsystem**

The INTHW CI includes an Interface Server. Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the Interface HWCI and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping. The Interface Server supports the Client Subsystem and a portion of the Communications Subsystem. DMS software runs on this host: DDICT and V0 GTWAY. The Data Dictionary Server (EcDmDictServer) allows authorized users to perform data searches, inserts, updates and deletions to data within the Data Dictionary Database. The V0 GTWAY consists of multiple processes to allow access to data and services between the ECS Data Server and the V0 IMS System.